UNIVERSIDADE DE CABO VERDE

Vera Lúcia Ramos Cordeiro Mendes Teixeira

Relatório do Projecto do Estágio

SISTEMA DE GESTÃO UNIVERSITÁRIA

IMPLEMENTAÇÃO E PARAMETRIZAÇÃO DO MÓDULO

RECURSOS HUMANOS

LICENCIATURA EM TECNOLOGIAS DE INFORMAÇÃO E COMUNICAÇÃO

CAMPUS DE PALMAREJO – CIDADE DA PRAIA

Dezembro, 2009

Vera Lúcia Ramos Cordeiro Mendes Teixeira

Aluna nº – 2812

Sistema de Gestão Universitária

Implementação e Parametrização do Módulo Recursos Humanos

Relatório do Projecto do Estágio Realizado nos Serviços Técnicos

Relatório do Estagio apresentado como requisito parcial para obtenção do grau de Licenciatura em Tecnologias de Informação e Comunicação, sob a orientação do Eng.º Armindo Castelo Branco e Co-Orientação do Celestino Lopes Barros

AGRADECIMENTOS

Ao longo destes meses em que estive envolvido no estágio da licenciatura em Tecnologias de Informação e Comunicação e na, respectiva, preparação e redacção deste documento, tive a oportunidade de contar com o apoio de diversas pessoas que, directa ou indirectamente, contribuíram para a obtenção do presente relatório.

Agradeço à Deus pela grande força, que me acolheu em todas as horas de minha vida.

Aos meus orientadores, Eng. Armindo Castelo Branco e o Co-orientador Dr. Celestino Barros, meus agradecimentos especiais, pela valiosa orientação, senso crítico e fundamental contribuição ao longo da realização deste trabalho.

Agradeço meus colegas de equipa de implementação do sistema, Flávio, Olavo, Sandra, Frutuoso, Orlanda e Mónica, pelo constante apoio e por partilharem comigo as conquistas e as frustrações. Muito obrigado.

Agradeço, ainda, todos aqueles que, com o seu saber, a sua colaboração e o seu apoio crítico, dispuseram do seu tempo para debater comigo orientações e práticas de implementação de sistema no contexto da Universidade de Cabo Verde.

DEDICATÓRIA

Este trabalho é dedicado à minha Família, em especial ao meu filho Sandro, por terem de suportar a minha ausência e mau humor durante o tempo em que esteve envolvido nesse curso.

"Feliz aquele que transmite o que sabe e aprende o que ensina."

(Cora Coralina, 1889-1985)

Lista de Siglas

GNU - General Public License

PHP - Hipertext PreProcessor

GTK - GNU Image Manipulation Program ToolKit

HTML - Hiper Text Markup Language

TI – Tecnologia de Informação

SGBD - Sistemas de Gestão de Dados

SAGU - Sistema Aberta de Gestão Unificada

SGU - Sistema de Gestão Universitária

SI - Sistema de Informação

OSS - Open Source Sofware

GPL - GNU General Public Licence

POO - Programação Orientada a Objectos

RUP - Rational Unified Process

URL - Universal Resource Locator

Índice

Índice de Figuras

Índice de Tabelas

Resumo

O presente relatório, apresenta a implementação de um módulo do Sistema de Gestão Universitária: Módulo de Recursos Humanos. O estágio, cuja duração foi de 265 horas, foi realizado na Universidade de Cabo Verde, no Departamento de Ciências e Tecnologias – Serviços Técnicos, Campus Palmarejo. O estágio decorreu entre os meses de Julho a Dezembro, sob a orientação da Engº Armindo Castelo Branco e co-orientação do Dr. Celestino Barros.

O objectivo do sistema (Módulo Recursos Humanos) é facilitar o acesso à documentação e automatizar o processo de oficialização dos mesmos, usando para isso o ambiente Web, sendo acessível via *browser* nas estações através da intranet. A modelagem do sistema foi feita utilizando notação UML. O sistema foi desenvolvido em PHP, com bases de dados *PostgreSQL.*

Assim, neste relatório apresentam-se algumas considerações, gerais, sob o ponto de vista teórico, sobre o mundo da informação. Caracteriza-se a estrutura orgânica funcional da Uni-CV, com especial incidência para o Departamento de Ciências Tecnologias, mais especificamente para o Departamento de TIC, da qual faz parte integrante a implementação do Sistema de Gestão Universitária. Descreve-se, igualmente, as actividades levadas a cabo pela estagiária aquando do seu estágio na implementação do sistema. Por último, procura-se tecer algumas posições críticas relativas ao papel dos estágios e sua praticabilidade no contexto universitário.

Para o desenvolvimento do projecto de estágio fui integrado numa equipa composta por oito elementos, sendo sete alunos estagiários, que concentram esforços no desenvolvimento dos módulos que compõem o projecto SGU em desenvolvido, na universidade.

Durante o estágio desempenhei fundamentalmente funções de análise e programação do módulo de Recursos Humanos deste sistema.

Palavra-Chave – Sistema de Gestão Integrada, Tecnologias *Open source*

I. INTRODUÇÃO

1.1. Contextualização

Criada como um dos principais centros de educação e cultura da sociedade cabo-verdiana, a Universidade de Cabo Verde (Uni-CV) apresenta-se como Instituição Publica cuja função é o desenvolvimento de novas gerações de profissionais, cidadãos, capazes de contribuir para o desenvolvimento do País e o crescimento pessoal. Condições para uma formação académica que atenda a estes requisitos exigem que os conteúdos programáticos e sua dinâmica de aprendizagem sejam estruturados com carácter de formação, fornecendo a seus estudantes não somente os elementos cognitivos do processo ensino - aprendizagem, mas também, formação teórico-prático e valores culturais, no sentido geral.

Assim, na demanda da excelência em educação através de uma gestão eficiente das informações, a Universidade de Cabo Verde, implementou um Sistema de Gestão voltado, para as actividades académicas, administrativas e financeiras.

Este Sistema de Gestão traduz-se num projecto denominado SGU, um sistema completo de gestão administrativo de Instituições de Ensino Superior. O sistema é independente a nível da plataforma, tanto em hardware como em software, pois foi desenvolvido com base numa tecnologia Web. Seu objectivo é integrar todos os departamentos administrativos, tais como Serviços Académicos, Recursos Humanos e Contabilidade.

A tecnologia e ferramenta utilizada para o desenvolvimento do referido sistema, consiste no uso da linguagem PHP, bases de dados *PostgreSQL* e Servidor *Web Apache* num ambiente GNU/Linux, cuja licença de utilização é GPL - *General Public License* (*Software* Livre). As principais vantagens do projecto para a Uni-CV são: custos relativamente baixo no desenvolvimento e flexibilidade, tendo em conta a forma como o sistema foi desenvolvido e implementado.

A minha integração, neste projecto, enquadra-se no âmbito da exigência da Uni-CV – Universidade de Cabo Verde, para efeitos de obtenção do grau de Licenciatura em Tecnologia de Informação e Comunicação.

Esta experiência representou para a equipa um enorme desafio, pois exigiu um trabalho árduo, que implicou muita pesquisa, e recurso a apontamentos de outras cadeiras/disciplinas frequentadas em anos anteriores.

Através deste projecto pretende-se mostrar que uma boa coordenação de esforços entre várias actividades, pode criar sinergias e consequentemente levar a um melhor aproveitamento dos recursos. Assim sendo, para a sua elaboração foi feita uma pesquisa para aproximarmos ao máximo da realidade, utilizando para o efeito dados da Internet, livros e consulta de trabalhos dos anos anteriores.

O Sistema foi concretizado, trazendo valor acrescentado para o Cliente, a Uni-CV, com melhorias substanciais na organização, de forma eficiente, de todo o suporte documental necessário. Pois, valoriza e acompanha os princípios da instituição, na medida em que procura melhorar soluções, no sentido de acrescentar valor à sua oferta e principalmente aos seus clientes e utentes/discentes.

1.2. Caracterização da Universidade

A Universidade de Cabo Verde teve a sua origem no ano de 1979, na então Escola de Formação de Professores que mais tarde evoluiu para o Instituto Superior de Educação. O objectivo fundamental desta Instituição era formar professores para o Ensino Secundário.

Por outro lado em 1984, foi formalmente criado o Centro de Formação Náutica, projecto apoiado pela cooperação Norueguesa. A instituição tinha como missão formar pessoal do MAR, a todos os níveis, e promover a investigação no domínio das Ciências e Tecnologias

O aumento da procura do ensino superior, associado a novas exigências de uma sociedade mais reivindicadora, tem levado o país a dinâmicas adaptativas do seu modelo do ensino superior, caracterizado pela reorganização das instituições públicas até então existentes.

Neste contexto, em 2004 foi criada através do Decreto-lei nº 31/2004, a Comissão Nacional para a Instalação da Universidade de Cabo Verde (CNI-UniCV), tendo por missão programar e desenvolver as actividades necessárias à instalação da Universidade Pública de Cabo Verde.

Em 21 de Novembro de 2006, instala-se realmente a Universidade de Cabo Verde visando contribuir, de forma decisiva, para o desenvolvimento durável e sustentável de Cabo Verde nos domínios científico, tecnológico, económico, social e cultural.

Esses três institutos associados ao ENG – Escola de Negócios e Governação vão constituir a mola mestra da Uni-CV, instituição pública, sem fins lucrativos, sediada na Cidade da Praia.

1.2.1. Organigrama da Universidade

O Conselho da Universidade de Cabo Verde actualmente é constituído por Reitor, Administrador Geral, Conselho Administrativo, Conselho Estratégico e por Conselho de Qualidade. Na Estrutura encontra-se vários Departamentos, Centros, Serviços Académicos, Serviços Técnicos, Acção Social, Financeiro Administrativo e entre outros. Tudo isso constitui a orgânica da Uni-CV apresentado na figura 1.

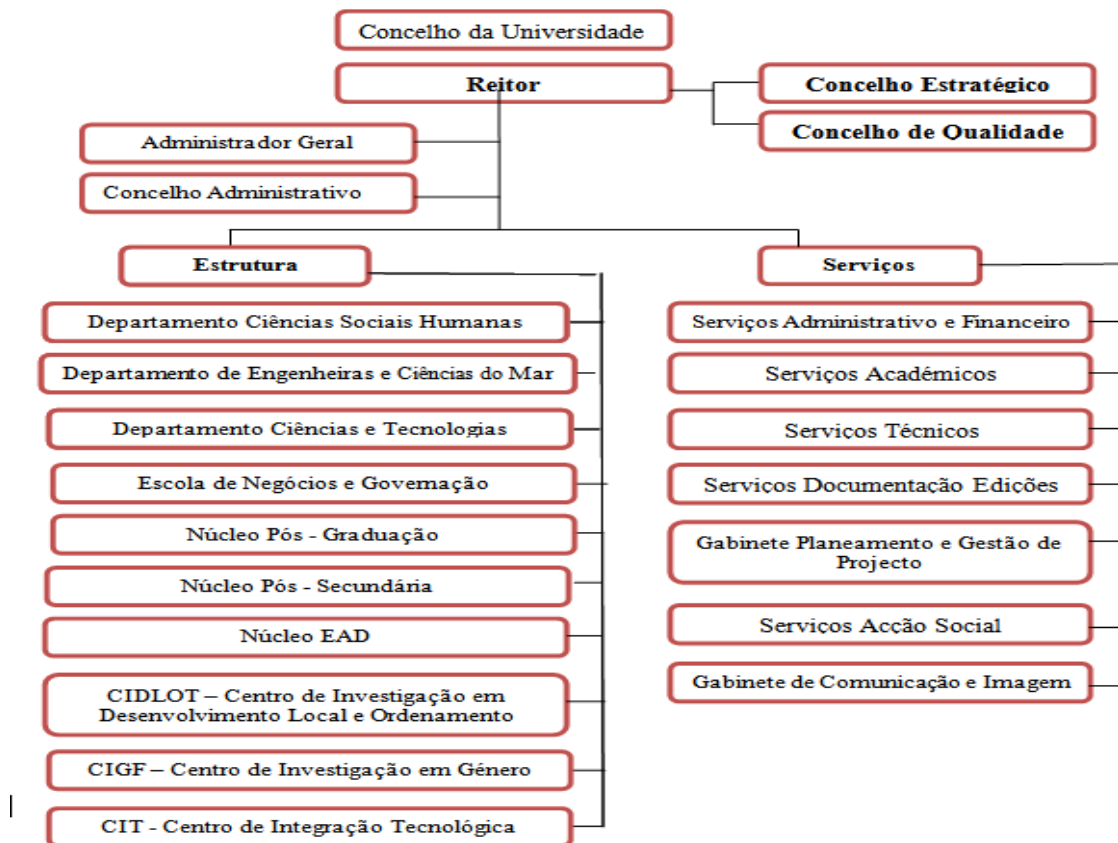



Figura 1: Organigrama da Uni-CV

Fonte: Marques 2009

1.3. Objectivos

1.3.1. Objectivo Geral

O objectivo deste relatório consiste em descrever o desenvolvimento e implementação de um sistema Web (Sistema de Gestão Universitária) para informatização da documentação (contrato) realizado durante o estágio. Com a implementação do sistema vai ser possível disponibilizar documentos através da intranet e da internet, eliminando assim a necessidade de versões impressas dos documentos e obtenção de um melhor controlo e organização dos mesmos.

1.3.2. Objectivo Específicos

- Descrever a implementação da gestão de Recursos Humanos na Universidade;
- Reconhecer os principais benefícios que o sistema de gestão proporciona à Universidade.
- Diminuir a necessidade de manipulação de documentos impressos
- Obter um controlo mais específico sobre o acesso dos utilizadores a cada tipo de documento, de acordo com o departamento a que pertencem
- Modelar o sistema utilizando a linguagem UML;
- Modelar a arquitectura do sistema;
- Desenvolver o Módulo Recursos Humanos;
- Garantir a segurança de informação.

1.4. Organização do Trabalho

O presente relatório encontra-se dividido em 4 (quatro) capítulos. No primeiro uma introdução que nos permite enquadrar e contextualizar o projecto desenvolvido; no segundo é apresentado a fundamentação teórica, descrevendo o SAGU, (Sistema Aberto de Gestão Unificada) suas características, seus módulos, história e tecnologias *open source*, onde é possível perceber que necessitamos de tecnologias *open source*; no terceiro capítulo é descrito o processo de implementação do sistema, revelando os seus benefícios para Universidade; e, finalmente no quarto capítulo, é apresentado as conclusões, propostas para desenvolvimentos futuros do projecto descrito e apreciação final; também apresentamos uma lista das bibliografias referenciadas e os anexos.

1.4.1. Metodologia

A metodologia, segundo Teixeira (2004), quer dizer o "*estudo dos caminhos, dos instrumentos utilizados para se fazer pesquisas científicas, os quais respondem como fazê-lo de forma eficiente.*"

Para a realização deste trabalho surgiram-se duas estratégias complementares. De um lado, a pesquisa bibliográfica que foi realizada, em livros, artigos científicos, Internet, sebentas, monografias e relatórios visando a consistência teórica do mesmo e o levantamento de informações através de argumentos de autores, que permitiu abrir horizontes e tomar conhecimento de experiências similares ou relevantes.

E, doutro lado, a realização do estágio na organização para a implementação do sistema, onde foi possível efectuar encontros de trabalho com o responsável dos estagiários, e dos órgãos da Universidade.

Fomentou-se também uma troca de experiência, entre colegas estagiários.

1.4.2. Descrição de Problema/Justificativa

No mundo hodierno, globalizado, a grande rapidez com que a informação é disseminada pressupõe pessoas e instituições sempre actualizadas em seus conhecimentos.

No contexto dos órgãos públicos e, especificamente, na Universidade de Cabo Verde, onde se encontra uma abundante produção de informação, esse cenário não é diferente. A Universidade de Cabo Verde precisa ficar atenta a essas novas premissas da era tecnológica.

Essa nova realidade infere a necessidade de disponibilizar informações actuais e em tempo real. E, neste sentido, a optimização dos processos de organização e recuperação da informação nesta unidade, através da mudança de software de automação, que pudesse acelerar o processo de circulação de informação com a qualidade requerida pelos utilizadores da mesma faz-se necessário.

DAMÁSIO (2007) utiliza como justificativa para a utilização desse tipo de software a filosofia de sua utilização (sem custos de licença), com desenvolvimento constante e partilha, onde, dependendo do tipo de software pode ser desenvolvido a nível internacional.

Com a expansão e a sua transformação em Uni-CV, um sistema de gestão com as características acima torna-se uma necessidade preeminente, tendo em atenção a dispersão dos campus e da documentação.

Face a este cenário, foi tomada pelo responsável dos serviços técnicos e os demais órgãos competentes da Uni-CV, a decisão de procurar uma solução que pudesse ser desenvolvida internamente, de modo que a solução que viesse a ser implementado reflectisse, efectivamente, o modo de funcionamento desejado para os serviços académico, financeiro e administrativo na universidade, de acordo com a sua especificidade organizacional e estrutural.

II. FUNDAMENTAÇÃO TEÓRICA

2.1. Sistema de Informação nas Universidades

Actualmente os SI/TI (Sistema de Informação/Tecnologias de Informação) têm um papel estratégico na condução das organizações, chegando mesmo a modelar a própria estrutura institucional.

KARADIMA (1987), destaca que a importância dos sistemas de informações para as instituições universitárias "justifica-se na medida em que estas apresentam uma grande quantidade e complexidade de dados, bem como pela necessidade de procedimentos de tratamento da informação mais precisos e rápidos". Entretanto, a mais forte justificativa está na capacidade dos sistemas de apoiar o processo de tomada de decisão.

O mesmo autor apresenta como objectivos mais importantes dos sistemas de informações administrativas nas instituições de ensino superior:

- Responder às necessidades de informação externa e interna da instituição, apoiando a tomada de decisões académicas e administrativas;

- Eliminar a duplicidade de dados e deduzir o trabalho redundante em sua manipulação; Processar e analisar uma grande quantidade de dados com maior velocidade e flexibilidade no seu tratamento, de modo a apoiar significativamente o processo de tomada de decisões;

- Distribuir informações de melhor qualidade a todas a divisões e níveis administrativos da universidade; Ter acesso rápido e efectivo aos centros de excelência universitária no mundo, atingindo, com isso, maior produtividade e menor custo.

WOLYNEC E MARIN (1988) defendem a importância de centralizar as informações em um único sistema, isto porque, na maioria das universidades, a informação existente está espalhada em diferentes departamentos e estruturas administrativas, e barreiras departamentais impendem o fluxo das informações. "Muitas vezes a informação necessária à tomada de decisão ou à avaliação envolve dados de várias unidades, necessitando, dessa

forma, ser coordenada, integrada e central". A proposta dos autores é, portanto, integrar a organização universitária através de um sistema de informações também integrado.

2.2. A Contribuição dos Sistemas de Informações na Gestão Universitária.

Segundo KARADIMA (1987), o executivo precisa de subsídios que o ajudem no processo decisório. Esses subsídios devem possibilitar a identificação das características do problema, do tema ou assunto que requer decisões. Devem também identificar o contexto ou o ambiente relacionado às decisões e os possíveis impactos causados pelas decisões tomadas.

O grande desafio que os administradores enfrentam nos dias actuais, é o de prever os problemas e conceber soluções práticas a eles, a fim de realizar os anseios objectivados pela organização. De acordo com TEIXEIRA (2006), os administradores precisam estar muito bem informados, pois a informação é a base para toda e qualquer tomada de decisão. Os sistemas de informação têm um papel fundamental e cada vez maior em todas as organizações de negócios, pois quando eficazes, podem ter um impacto enorme na estratégia corporativa e no sucesso organizacional.

Para PEREIRA (1999) a utilização da informação e, sobretudo, dos sistemas de informação no processo de administração estratégica é crucial, visto que, os mesmos insistem em utilizar canais informais para obtenção de vantagens e para a tomada de decisão. Assim, a clientela é visto como uma forma de se obter benefícios para as unidades. E ainda, a importância da informação no processo de administração estratégica dá-se, sobretudo, na análise ambiental e no acompanhamento estratégico.

Os sistemas de informação são, portanto, ferramentas úteis à gestão universitária. Entretanto, é importante observar que não são os computadores, os sistemas, ou mesmo a disponibilidade da informação que resolvem os problemas da organização, mais sim a forma como ela irá ser aproveitado.

2.3. Software Open Source

Um sistema operativo, como outros programas, pode ser software proprietário (por exemplo Microsoft) ou de código aberto (por exemplo Linux). O software proprietário é comercial, ou seja, um utilizador ao adquiri-lo precisa de comprar uma licença de utilização e não pode fazer cópias do programa para terceiros. O software de código aberto, que pode ser comercial ou não, permite a visualização do código-fonte e é utilizado sob uma licença GPL (General Public License). Esta licença permite que o software seja distribuído e comercializado por qualquer pessoa, desde que essa avise o receptor acerca dos termos da licença. Permite ainda que qualquer pacote de software que deriva de outro protegido pela GPL, seja também abrangido por essa licença, bem como todo o código-fonte do programa protegido esteja acessível publicamente. Ou seja, deve ser fornecido aos utilizadores ou informada a maneira de o obter.

A importância do acesso ao código-fonte concentra-se em ser adaptável às necessidades de cada utilizador.

O software de código aberto teve origem na sociedade GNU (GNU is not UNIX) que derivou da *Free Software Foundation* (FSF), que estava descontente por o sistema operativo UNIX estar cada vez mais comercial e fechado.

Segundo a GNL- *General Public License,* o termo software livre se refere à liberdade do utilizador em executar, copiar, distribuir, estudar, modificar e aperfeiçoar o software. Destaca-se entre essas:

- A liberdade de executar o programa, para qualquer fim;
- A liberdade de estudar o funcionamento do software, e adaptá-lo para as necessidades. Acesso ao *source-code* é um pré-requisito para essa liberdade;
- A liberdade de redistribuir cópias de modo que possa ajudar ao seu próximo;
- A liberdade de aperfeiçoar o programa, e liberdade aos aperfeiçoamento, de modo que toda a comunidade se beneficie. O acesso ao fonte também é um requisito para esta liberdade.

2.3.1. Ubuntu

Pelo facto de existirem muitas distribuições do *linux*, torna-se essencial um estudo das vantagens e desvantagens de cada uma destas, bem como das aplicações às quais cada uma melhor se adapta.

O *Ubuntu* se destaca por ser um sistema que suporta a instalação do SAGU, pela facilidade de instalação, utilização e uma automatização das funções, uma interface bastante intuitiva, características que são bastante vantajosas sobretudo para quem esteja acostumado a sistemas operativos Windows.

O *Ubuntu* é um sistema operativo (S.O.), com um conjunto de programas que permitem o seu funcionamento perfeito. Este sistema autónomo baseia se na distribuição espontânea do conhecimento, os programas são preferencialmente livres (*free software*) e de fonte aberta (*open source*), o que significa que quem souber e estiver interessado, pode obter e alterar qualquer programa de acordo com as suas necessidades, tendo apenas a obrigação moral de o "mostrar", tornar disponível o "seu" programa para que outros também o possam desenvolver em função das suas carências. (TEIXEIRA & GOMES, 2006)

O *Ubuntu* usa uma licença GPL (*General Public License*), que permite a sua utilização livre e gratuita.

2.3.2. Miolo Framework

De acordo com BROD (2004), Miolo é um Framework escrito em PHP, utilizando apenas conceitos de POO (Programação Orientada a Objectos), que oferece uma grande gama de funções para optimizar e agilizar o processo de criação de sistemas.

Segundo ele, o MIOLO é um *framework* voltado para o desenvolvimento de sistemas WEB e foi desenvolvido na linguagem PHP, sendo resultado da evolução da metodologia de desenvolvimento de sistemas em *software* livre da Solis.[1] A sua arquitectura é adequada

[1] Solis, Cooperativa de Soluções Livres, é uma cooperativa de serviços formada no início de 2003, com o apoio da Univates, Centro Universitário, em Lajeado http://sagu.solis.coop.br/

para o desenvolvimento de aplicações acessíveis via *Web*, pois, utiliza o php5, que é uma poderosa linguagem de script.

Funcionalidades básicas

Visando facilitar a criação de módulos através do *framework* MIOLO, algumas funcionalidades básicas são disponibilizadas. Essas funcionalidades são ortogonais aos módulos, ou seja, todo módulo herda automaticamente as funcionalidades oferecidas pelo *framework*. Assim temos:

- Autenticação de utilizador;
- Controlo de permissão de acesso;
- Camada de abstracção para acesso a bases de dados;
- Gerir sessões e manutenção de *log*;
- Tratamento da página como um *webform*, com captura de eventos;
- Validação de entrada em formulários;
- Parametrização de layout e temas;
- Geração de arquivos em PDF.

2.3.3. Servidor Apache

O Apache é o servidor *Web*[2] mais popular do mundo. É um sistema gratuito e está presente em quase todas as distribuições *Linux* (AMARAL, 2009).

[2] Servidor web é um computador que contem um programa especial para "servir" páginas web, isto é, quando se visita uma página web o que se está na realidade a fazer é "pedir" páginas a esse servidor. http://www.icewarp.com.br/Products/web_server/index.php

Segundo TEIXEIRA (2006) o servidor *apache* é de entre todos os servidores o mais usado devidas as suas principais características, grandemente configurável, pode ser executado em diferentes plataformas, é flexível, está sempre em desenvolvimento para a inclusão dos protocolos mais actualizados (por exemplo, HTTP), fornece o código fonte completo e não possui licenças restritivas, pode ser configurado para diferentes funções, é composto de módulos, cada um implementando uma característica diferente e aumentando a funcionalidade do servidor, além de várias outras características.

Além disso, o Apache é uma entidade orgânica, aqueles que beneficiam dele usando-o, por outro lado, contribuem fornecendo realces das suas características e melhorando-o. A quantidade de esforço dispendido por todos os que o usam na sua melhoria é grande o que faz com que o produto resultante seja muito forte, isto acontece somente com *freeware* (grátis) porque quando alguém paga pelo software não está geralmente dispostos a reparar erros.

Abaixo estão algumas características que fazem com que este servidor Web seja o preferido, segundo TEIXEIRA (2006):

- Suporte a autorização de acesso podendo ser especificadas restrições de acesso separadamente para cada endereço/arquivo/directório acedido no servidor;
- Autenticação requerendo um nome de utilizador e senha válidos para acesso a alguma página/sub-directório/arquivo (suportando criptografia);
- Negociação de conteúdo, permitindo a exibição da página Web no idioma requisitado pelo Cliente Navegador;
- Suporte a virtual *hosting* (é possível servir duas ou mais páginas com endereços/ portas diferentes através do mesmo processo ou usar mais de um processo para controlar mais de um endereço);
- Suporte a IP virtual *hosting;*
- Suporte a *name* virtual *hosting;*
- Suporte a servidor Proxy FTP e HTTP, com limite de acesso;

- Suporte a proxy e redireccionamentos baseados em URLs para endereços Internos *(Foundation*, 2007.

2.3.4. Base de Dados PostgreSQL

- O PostgreSQL é um SGBDOR (Sistema de Gestão de Bases de Dados Objecto Relacional) *open-source* que vem sendo mantido a mais de 15 anos. O PostgreSQL é actualmente o mais avançado SGBD de código aberto. Ele possui características relacionais e também características de orientação a objectos tais como herança e tipos de dados personalizados POSTGRESQLBRASIL, (2007). O PostgreSQL utiliza o modelo cliente-servidor, consistindo em 2 processos. Um servidor que gere a base de dados, conexões e executa acções no banco de dados. E um cliente que requer acções sobre a base de dados, enviando-as para o servidor (POSTGRESQLBRASIL, 2007).

- Entre as vantagens de seu uso destacam-se um melhor suporte que SGBD's proprietários, extremamente confiável e estável, extensível, multi-plataforma e é projectado para suportar ambientes de grande volume de dados.

2.3.5. PGADmin III

O PgAdmin III é uma poderosa plataforma de desenvolvimento para bases de dados PostgreSQL, muito utilizado devido não só a todas as suas qualidades como por se tratar de um programa freeware e ainda por correr quer em Windows 2000/XP quer em ambientes GNU/ Linux. O PgAdmin III foi desenhado de forma a responder às necessidades de todos os seus utilizadores, quer desde simples query's em SQL até ao desenvolvimento de bases de dados complexas. A interface gráfica está feita de forma a suportar todas as potencialidades do Postgres e facilitar a sua administração. (TEIXEIRA, 2006)

A aplicação possui ainda um construtor de query's, um editor de SQL, um editor de código server-side, afirma o mesmo autor.

É uma ferramenta bastante simples, acessível e intuitiva.

2.3.6. PHP

O PHP é uma linguagem de criação de scripts, cujo código é inserido dentro do código HTML, e processado no servidor, sendo retornado para o cliente apenas HTML KINOSHITA, (2001).

O PHP é uma linguagem de programação de computadores interpretada, livre e muito utilizada para gerar conteúdo dinâmico na Web. Apesar de ser uma linguagem de fácil aprendizagem e de uso para pequenos scripts dinâmicos e simples, o PHP é uma linguagem poderosa orientada à objectos. TEIXEIRA, (2006).

Este produto pode ser adquirido através do site oficial: http://www.php.net, sem custo de licença.

Características do PHP

As características da linguagem de programação PHP, foram se desenvolvendo e sucedendo a medida que novas versões da linguagem foram surgindo, segundo as afirmações de HUDSON (2005), na abordagem à história da linguagem.

O PHP é a linguagem mais usada em aplicações Web em todo o mundo. Este privilégio deve-se às suas importâncias das quais, algumas são referidas por BUCHANAN (2002):

- Object-Oriented (Orientado a Objectos (OO)). PHP não é totalmente uma linguagem orientada a objectos mas suporta o conceito OO;

- Cross-Plataform (Independente da plataforma). As maiores linguagens server-sides incluindo ASP e JSP têm foco em certas plataformas WWW. ASP por exemplo em IIS da Microsoft. O PHP foi desenhado para funcionar em diferentes tipos de plataformas WWW e hardware. Pode ser facilmente integrado com vários tipos de Web Server.

- Integração com Bases de Dados. PHP pode ser integrado com os sistemas de bases de dados mais populares, incluindo MySQL, Oracle ou SyBase.

- Actualização das versões. A linguagem é constantemente actualizada com novas versões. O PHP é uma linguagem dinâmica que pode rapidamente ser mudado, o que é muito bom para a sua promoção.

- Realce à segurança. A linguagem oculta o código do utilizador, permitindo a estes ver apenas o código processado em HTML.

- Realce às ferramentas de desenvolvimento. A linguagem tem um leque completo de ferramentas de desenvolvimento, que permite a criação de testes de código PHP.

2.3.7. Agata Report

Agata Report é um gerador de relatórios que funciona tanto no Linux, quanto no Windows.

É capaz de conectar a vários bases de dados, como, por exemplo, PostgreSQL, MySQL, SQLite, Oracle, DB2, MS-SQL, Informix, InterBase, Sybase, ou Frontbase.

Permite exportar os relatórios em formatos como PostScript, plain text, HTML, XML, PDF ou CSV (StarCalc, Excel), além de gerar gráficos.

Com ela, o utilizador pode definir níveis de dados, subtotais e grande total para o relatório. Permite, também, criar documentos, como cartas e mesclar dinamicamente com os dados provenientes do banco de dados. O *Agata Report* também pode criar etiquetas de endereçamento e até gerar o diagrama ER completo a partir de seu banco de dados.

2.3.8. UML (Unified Modeling Language)

Segundo GUEDES (2003), a UML (*Unified Modeling Language* ou Linguagem de Modelagem Unificada) é uma linguagem visual utilizada para modelar sistemas computacionais por meio do paradigma de Orientação a Objectos. Essa linguagem tornou-se, nos últimos anos, a linguagem padrão de modelagem adoptada internacionalmente pela indústria de Engenharia de *Software*.

A modelagem de sistemas torna-se necessária, uma vez que existe uma alta complexidade do mesmo. Deste modo, a UML torna-se ideal para realizar essa tarefa, pois é capaz de abranger diferentes visões voltadas à arquitectura do sistema (BOOCH, 2000).

Porém foram utilizados para o desenvolvimento deste projecto os seguintes diagramas: o Diagrama de Casos de Uso, o Diagrama de Classes e o Diagrama de Actividades

Este capítulo abordou as tecnologias de suporte a aplicações orientadas para a Web, onde se tentou justificar o porquê da utilização de umas em detrimento de outras.

A estrutura de suporte às aplicações aqui apresentadas consiste num plataforma Linux com servidor apache e base de dados PostgreSql.

Essa estrutura teve influência decisiva na selecção da linguagem utilizada para o desenvolvimento das aplicações descritas

2.4. Sistema Aberto de Gestão Unificada (SAGU)

Segundo SOLIS (2009) SAGU - Sistema Aberto de Gestão Unificada, é um sistema capaz de gerir todo o relacionamento de um aluno com uma instituição de ensino, dentro da esfera académica, administrativa ou financeira.

O SAGU é constituído por módulos (processo selectivo, académico, financeiro, contabilidade e Recursos Humanos) que podem ser estruturados para atender às necessidades dos mais diversos tipos de instituições de ensino.

Ainda a mesma fonte afirma que o SAGU é um sistema em constante desenvolvimento, e é construído apenas com ferramentas de software livre, como o sistema operativo GNU/Linux, a linguagem de programação PHP e a gestão de bases de dados *PostgreSQL.* O SAGU é também um Software Livre, distribuído sob a licença GPL.

2.4.1. Principais Características do SAGU

Relativamente ao acesso às informações o SAGU, tem um fácil acesso, uma interface 100% WEB, moderna e intuitiva, onde todas as informações são disponibilizadas de forma prática e acessível. Quanto ao desempenho o SAGU proporciona rapidez de acesso mesmo em redes de alto tráfego e computadores obsoletos. No que diz respeito a segurança, é um sistema que faz a gestão de utilizadores, criando políticas de permissão e restrição de acessos. E ainda proporciona custo zero de licença de software, pois utiliza-se ferramentas livres.

- **Independência de bases de dados**

Como as dificuldades que defrontavam no sistema antigo eram, em grande parte, devidos à amarração que tinham com uma base de dados proprietária, o novo sistema não lhes permitiam a acessível migração para qualquer outra base caso isto se apontasse necessário.

- **Interface Web**

Os utilizadores deveriam ter acesso ao sistema, independente do sistema operativo que executavam em seus computadores, e o novo sistema também deveria possibilitar que no futuro migrassem as estações dos clientes também para software livre (o que começou a ser feito em Janeiro de 2000). A melhor maneira de se conseguir isto era utilizando um browser padrão (Netscape ou Internet Explorer) como cliente.

- **Modularidade**

O sistema seria decomposto em módulos específicos e bem determinados, de forma que o processo de desenvolvimento pode ser melhor dividido entre os membros da equipa e facilitar a colaboração de outros.

Acesso ao código fonte

O SAGU está sob licença GPL, ou seja, seu código fonte é distribuído de forma gratuita.

Base de dados robusta

O SAGU trabalha com base de dados *PostgreSQL*, que é um poderoso sistema de base de dados relacional de código aberto.

O SAGU2 foi desenvolvido utilizando o Miolo, um *framework* para desenvolvimento em Software Livre. O aproveitamento do mesmo permite uma maior flexibilidade no progresso do sistema, pois actua com conceitos de orientação a objectos, possibilitando assim o reaproveitamento de código. O Miolo permite ainda uma maior modularização do sistema.

- **Custo zero de licença de software**

O SAGU foi criado para trabalhar com custo zero de licença de software. No servidor, é utilizado o sistema operativo Gnu/Linux e a base de dados *PostgreSQL* A linguagem de programação é PHP, e o *framework* de desenvolvimento é o Miolo. Todos sob licença GPL, ou seja, de livre distribuição e cópia. O acesso ao sistema pode ser feito através de qualquer sistema operativo, contanto que possua um navegador de internet, o que torna o SAGU um sistema multiplataforma.

2.4.2. Módulos

De acordo com SOLIS (2009), os principais módulos do SAGU versão 2 são:

- Básico
- Processo Selectivo
- Académico
- Financeiro
- Contabilidade
- Recursos Humanos
- Institucional
- Controlo de Cópias
- Questionário

Módulo Básico

O módulo Básico do SAGU2 é o módulo essencial para o funcionamento do sistema. Neste módulo existe a parte de configuração do sistema, onde é possível parametrizar o seu funcionamento de acordo com as reais necessidades, como, por exemplo, *logótipos*, cores, número de registos por listagens, métodos de autenticação, tamanho das janelas e campos dos formulários, formato de data e hora, número de casas decimais, mensagens de inclusão, alteração, exclusão e erros, e quais módulos estão instalados.

Além disso, nestes módulos são encontrados todos os registos básicos para o funcionamento e uso dos outros módulos, como o registo de pessoas, dados dos funcionários, professores, estudantes, instituições e empresas, dados geográficos, unidades e horários.

Módulo Académico

Este módulo gerência as informações académicas como cursos, disciplinas, currículos, períodos, matrículas, enfim, tudo o que é necessário para a montagem de uma grade de disciplinas para um aluno. Além disso, tem um rigoroso controlo desta grade curricular implementando controlo de versões, controlo de requisitos e liberação de disciplinas.

O módulo académico também possui um sub-módulo para o acesso dos alunos via Internet. Este faz consultas no módulo académico e mostra as informações que estão disponíveis para os alunos, como histórico escolar, salas de aula, horários, professores, etc. Além disso, este módulo tem a possibilidade de disponibilizar processos como a matrícula e os ajustes de matrícula pela Internet

Módulo Financeiro

O módulo financeiro consiste de um sistema de contas a receber, embora já esteja muito perto de um de contas a pagar também.

O módulo financeiro é consultado pelo módulo académico, quando os dois co-existirem, com o objectivo de integração de informações e assim evitar que um aluno que esteja inadimplemento e consiga matricular-se para o próximo período, por exemplo O sistema ainda possui uma série de relatórios como a listagem da dívida activa, previsões de lançamentos, de relatório com a possibilidade de impressão de mala directa de cobrança para estes alunos.

Módulo de Contabilidade

Este módulo, o controlo das contas contabilísticas e centros de custos da instituição permite através destes, a integração com os cursos. Ainda gera relatórios como a razão e livro-caixa.

Módulo Recursos Humanos

Módulo para gerir os salários dos funcionários, e professores. Contempla ainda a geração de remessas para folha de pagamento. É o início de um módulo completo de Recursos Humanos. Esse como é o nosso objecto de estudo retomaremos mais a frente.

Módulo Institucional

Através desse módulo é possível registar todos os recursos físicos (salas, laboratórios, auditórios) e materiais (computadores, projectores) da instituição. Com um controlo de versões é possível ter um histórico completo da evolução dos recursos da instituição.

III. MÓDULO RECURSOS HUMANOS DO SISTEMA

No capítulo anterior foram apresentadas algumas tecnologias, ferramentas e linguagem de programação para implementação do sistema. Neste capítulo descreveremos a implementação do módulo aplicacional de Recursos Humanos, recorrendo à linguagem de programação já apresentada.

3.1. Sistema de Gestão Universitária (SGU)

Sistema de Gestão Universitária (SGU), é um sistema desenvolvido num ambiente *Web*, possibilitando acesso via *browser*[3] (Internet *Explorer, Nestcape, Mozilla, Mosaic* etc) aos seus utilizadores. O sistema foi desenvolvido com todas as características herdadas do SAGU2. No Módulo de Recursos Humanos do SGU tem uma funcionalidade que é gestão de contrato que não é apresentado no SAGU.

3.1.1. Módulos do Sistema de Gestão Universitária (SGU),

Módulo Básico: É o módulo essencial para o funcionamento do sistema;

Módulo académico: Este módulo faz gestão das informações académicas, como cursos, matriculas, enfim, tudo o que é necessário saber acerca de um aluno;

Módulo Financeiro: Consiste em receber propinas.

Módulo Contabilidade: Possuí o controlo das contas contabilísticos da Universidade

Módulo Recursos humanos: Faz gestão da folha do salário, criação de contrato. Por ser o nosso objecto do estagio retomaremos esse tópico na secção ;

3.1.2. Funcionamento actual da secção de recursos humanos da Uni-CV

Segundo informações que obtivemos, através de entrevista formal, ao Dr. Iderlindo de Pina, responsável pela área de administração, ficamos a saber que todo trabalho de Recursos Humanos é feito por ele e pelo Sr. Salvador Moniz, responsável pela área financeira, manualmente.

---

[3] Software necessário para navegar na www (Word Wide Web).

3.1.3. Problemas identificados no sistema actual

Após a realização de entrevistas foram detectados os seguintes problemas no sistema actual:

- Controlo feito manualmente;
- Informações dispersas;
- Consoante o aumento de solicitações, o acesso a estes documentos tornam-se difíceis dificultado;
- Tempo de resposta da solicitação dificultado devido ao aumento da demanda de solicitações;
- Falta de segurança no armazenamento de documentos e informações: ex.: arquivo corrompido, acidente no local aonde são armazenados os documentos.

3.1.4. Módulos de Recursos Humanos

O objectivo do Módulo Recursos Humanos é atender as demandas no planeamento e gestão da Uni-CV, possibilitando uma gestão de documentação, mas concretamente gerir contratos.

Esse projecto visa a integrar informações heterogéneas, originarias de diferentes sectores da Uni-CV, por meio da construção de uma base de dados, a partir do qual técnicas de análise possam ser utilizadas para extrair informações de qualidade sobre a realidade da universidade, proporcionando a percepção rápida das mudanças e tendências, permitindo ao sector público uma melhor alocação dos recursos disponíveis.

Todas as funções necessárias para o funcionamento de base de dados no SGU foram implementadas em componentes de *software open source*, responsáveis por desempenhar tarefas específicas e interagir entre si, quando necessário.

Cada um desses componentes possui uma identificação única, conhecido por todos os outros componentes e utilizado para a comunicação entre eles, por meio dos protocolos de comunicação.

Para a implementação deste módulo levamos em consideração o modulo recursos humanos do SAGU

3.1.1. Razões Pela Escolha do SAGU

Segundo o Director dos Serviços Técnicos da Uni-CV (Prof, Celestino Barros), a gestão dos serviços da universidade deve basear-se em software livre foram analisados um conjunto delas conforme a tabela 1 e o SAGU principalmente por ser um sistema baseado em tecnologias *Open Source,* capaz de responder às necessidades de automatização das áreas Administrativas e financeira da Universidade, facilidade de utilização, interface intuitiva se destacou.

Com o objectivo de identificar os sistemas que melhor se adequa à Uni-CV foi feito um estudo comparativo entre seis *softwares*, todos implementados no Brasil, sendo dois deles livres conforme se descreve na tabela 1:

Tabela 1: Quadro Comparativo Entre Sistemas

Características	Sistemas					
	SAGU	**TesEscola**	**AdX**	**UniMestre**	**SophiA**	**Corpore RM Education al**
Por Categoria de Instituição de Ensino						
Ensino Básico	X	X		X	X	X
Escola Secundário	X	X		X	X	X
Ensino Superior	X	X	X	X	X	X
Por características funcionais						
Controlo Académico	X	X	X	X	X	X
Controlo Financeiro	X	X	X	X	X	X
Controlo de Processo Selectivo	X		X	X		
Recursos Humanos	X	X	X		X	X
Biblioteca	X	X	X	X	X	X
Por características operacionais						
Plataforma	Web	Desktop/ Web	Web	Desktop/ Web	Desktop/ Web	Desktop/ Web
Distribuição	Livre	Proprietário	Livre	Proprietário	Proprietário	Proprietário

3.1.2. Ferramentas para o desenvolvimento do módulo Recursos Humanos

Este projecto foi desenvolvido para fins universitária, para ser executado num ambiente Web. O desenvolvimento abrangeu recursos humanos, software e de *hardware* limitado sob as condições listadas abaixo:

Softwares para desenvolvimento:

- A linguagem utilizada para o desenvolvimento das páginas foi PHP;
- Apache: servidor Http;
- Servidor de Bases Dados foi o PostgreSQL;
- UML para modelagem de diagramas e do banco de dados;
- PgAdmin: administração das bases de dados;
- O servidor de páginas será executado no sistema operativo Linux;

Foi escolhido o sistema operativo Linux (*Ubuntu*), porque permite-nos aceder a um sistema operativo fiável, a programas com evolução constante e adaptados às necessidades de cada utilizador, sem depender dos preços estabelecidos pelas multinacionais do sector informático.

A linguagem PHP foi escolhida porque suporta várias plataformas e uma variedade de servidores Web, além de permitir conexões persistentes de base de dados.

O *PostgreSQL* foi escolhido pelas seguintes razões: Base de Dados relacional *Open Source*; altamente escalonável e fácil de administrar.

Foi escolhido o Servidor Web o Apache também pelo facto de ser *Open Sourcee* e o mais utilizado no mundo conforme se tinha dito anteriormente.

3.1.3. Principais Benefícios para Universidade

De forma geral, podemos dizer que o sistema proposto terá como principais benefícios:

- Rapidez no acesso as informações;

- Eliminação das principais deficiências do sistema actual;
- Personalização de acordo com os requisitos do cliente;
- Melhor gestão das informações;
- Custo considerado quase nulo;

Web Apache, o interpretador PHP, o SGBD e o sistema operativo Linux utilizando no servidor, são de distribuição gratuita. Ou seja, todos os recursos financeiros para o desenvolvimento foram direccionados à equipa de desenvolvimento. Além de aproveitar mão-de-obra mais acessível, pois a equipa de desenvolvimento/implementação foi constituído por estagiários do curso de Tecnologias de Informação e Comunicação sob orientação e supervisão de um Professor da Uni-CV.

3.1.4. Estrutura do Sistema Implementado

O SGU é um sistema modular, desenvolvido num ambiente *Web* (intranet e internet), e funciona cliente/servidor. Este tipo de arquitectura é a mais utilizada actualmente, pelo facto de ser a mais avançada, e o que mais evoluiu nos últimos anos.

Conforme podemos indicar a figura 2, existem inter-relacionamentos entre os módulos, permitindo que estes interajam através da solicitação de serviços. Desse modo, dependendo do nível das consultas, haverá necessidade de que diferentes módulos sejam activados.

A partir dessa estrutura, constata-se que o modelo de arquitectura de SGU proposto apresenta algumas características que lhe proporciona o potencial para se tornar uma ferramenta eficaz de apoio de gestão. A arquitectura cliente/servidor estabeleceu um novo paradigma de processamento de dados, diversificando a interoperabilidade entre o processo de software e o servidor. Além disso, a arquitectura proposta busca oferecer recursos que coordenem estes processos, de tal forma que a perda de sincronização não resulta em alterações ou perda de informações para o sistema.

Seu funcionamento se baseia no seguinte esquema: o utilizador do sistema, através do processo de software cliente, envia o pedido de requisição ao processo de software servidor, que por sua vez devolve ao cliente os resultados solicitados.

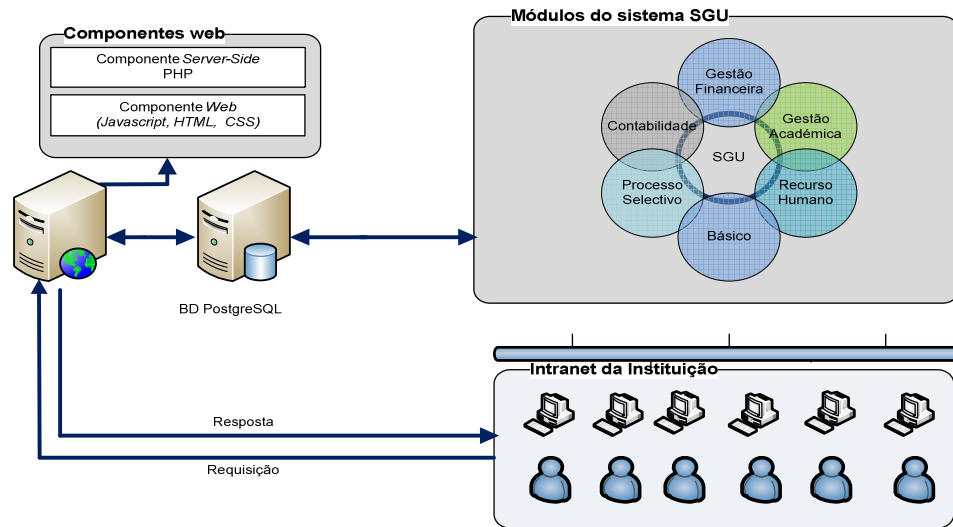



Figura 2: Estrutura do SGU

Fonte: Equipa de Implementação

3.1.5. Metodologias para a implementação do módulo Recursos Humanos

De acordo com AVISON e FITZGERALD, metodologia é o conjunto formado por procedimentos, técnicas, ferramentas e documentação que auxiliará os responsáveis pelo desenvolvimento de sistemas em seus esforços na implementação de um novo sistema de informação. (1997: p.10)

Uma metodologia consistirá de fases, cada uma consistindo de sub-fases, que orientarão estes responsáveis na escolha das técnicas que deverão ser mais apropriadas a cada estágio do projecto e também auxiliá-los a planear, gerir, controlar e avaliar o projecto do sistema de informação.

Existem várias metodologias de desenvolvimento disponíveis, como por exemplo:

- Convencional ou cascata;
- Incremental;
- Iterativo;
- Espiral;
- Prototipação;
- RUP;
- Extreme Programming (XP).

Vantagens da utilização de uma metodologia

- **Ganho de Produtividade**: É eficaz em definir o problema a ser resolvido, esclarecendo para toda equipa através da documentação gerada, todo o escopo da solução e seus aspectos relevantes.
- **Documentação**: Regista a memória do trabalho que está sendo desenvolvido, servindo para futuras revisões e implementações.
- **Padronização**: Ao utilizar os padrões citados na MDS, elimina-se os projectos em que só o fulano sabe como funciona. O conhecimento registado é entendido por qualquer outro técnico habilitado da Empresa.
- **Organização**: Seguir a metodologia substitui o processo artesanal e empírico de construção de sistemas.

Neste trabalho, estava previsto na ideia inicial, utilizar o Modelo Espiral, na modelagem do sistema. No entanto definimos a metodologia RUP (*Rational Unified Process*). O modelo espiral, por suas características de avaliação e planeamento baseadas em risco, exige que se tenha experiencias.

Em seu lugar utilizamos a abordagem simplificada da RUP pelo facto de ser totalmente customizável, contudo é dito como complexo, sendo recomendável para grandes equipas de desenvolvimento e a grandes projectos. KRUCHTEN, (2001).

O Processo Unificado proposto pela Rational (Rational Unified Process – RUP) foi criado para apoiar o desenvolvimento orientado a objectos, fornecendo uma forma sistemática para se obter reais vantagens no uso da Linguagem de Modelagem Unificada (Unified Modeling Language – UML).

Tendo em conta a grande complexidade do projecto, optamos por desenvolver o sistema em módulos, proporcionando um desenvolvimento mais simplificado, de forma, a libertar para uso institucional os módulos do sistema que são os seguintes: módulo básico, académico, financeiro, recursos humanos e módulo de contabilidade, sendo que todos eles interagem entre si de forma dinâmica e continua.

Além da divisão em módulos, o sistema reconhece vários tipos de utilizadores, que farão uso apenas ao módulo que possuem permissão de acesso.

O sistema fornece acesso a todos os módulos através de uma tela principal, que é accionada após o login do utilizador, como podemos ver na figura 2.

Fases do RUP

O ciclo de vida do projecto no RUP possui quatro fases: concepção, elaboração, construção e transição. Cada fase é concluída por um marco, ou seja, cada fase é basicamente um intervalo de tempo entre dois marcos. *:* **KRUCHTEN**, *(*2003 p. 272).

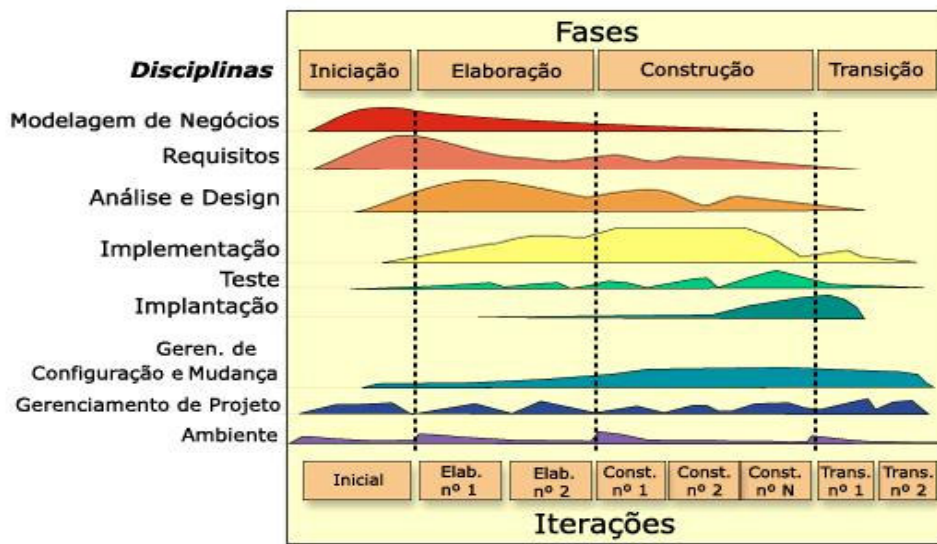



Figura 3: Ciclo de construção de uma versão usando RUP

Fonte: KRUCHTEN, 2003

3.2. Processo de implementação

O processo de implementação do software envolve adquirir conhecimentos sobre o software e sobre a plataforma operativos

Os requisitos básicos para instalação do *SGU* são: *Framework de desenvolvimento Miolo, PHP, PostgreSQL, Apache*. Também tivemos a necessária a verificação da forma do funcionamento das configurações interna do sistema, bases de dados, e testes com utilização dos diversos processos integrados ao sistema.

No servidor utilizamos o sistema operativo *Ubuntu/Linux*, base de dados *PostgreSQL*, estações de trabalho multiplataforma, linguagem de programação PHP e o *framework* de desenvolvimento Miolo, todos sob licença GPL, ou seja, de livre distribuição e cópia.

Trabalhamos com base de dados *PostgreSQL*, que segue padrões internacionais SQL e permite o registo de milhões de registos, sem restrições e limites.

3.2.1. Descrição do Situação Actual

Segundo informações que obtivemos, através de entrevista formal, ao Dr. Iderlindo de Pina, responsável pela área de administração, ficamos a saber que todo trabalho de Recursos Humanos é feito por ele e pelo Sr. Salvador Moniz, responsável pela área financeira, manualmente.

3.2.2. Problemas identificados no sistema actual

Com a realização da entrevista foram detectados os seguintes problemas no sistema actual:

- Controlo feito manualmente;
- Informações dispersas;
- Consoante o aumento de solicitações, o acesso a estes documentos tornam-se difíceis e dificultado;
- Tempo de resposta da solicitação dificultado devido ao aumento da demanda de solicitações;
- Falta de segurança no armazenamento de documentos e informações: ex.: arquivo corrompido, acidente no local aonde são armazenados os documentos.

3.2.3. Recolha e Analise de requisitos

A sobrevivência das organizações no competitivo mercado actual depende directamente da produtividade e qualidade da instituição. Isto significa dizer que o dinamismo e a crescente competitividade no mundo educacional colocam em risco a vida das instituições ao não questionar seus métodos tradicionais de gestão, desenvolvimento de novos produtos e serviços, produção e controlo da qualidade (GONÇALVES, 2007).

O levantamento dos requisitos foi efectuado pelo responsável do sistema e pelos estagiários junto aos gestores da Uni-CV, através de reunião para o evento, onde foi realizada a especificação da aplicação.

Levantamentos de dados

Foi escolhido a técnica de entrevista como técnica de levantamentos de dados a ser utilizado com o cliente, tendo como foco principal o funcionário responsável por toda a parte dos Recursos Humanos do campus palmarejo. Após a realização da entrevista foram levantados os seguintes dados sobre o processo actual realizado no sector.

Necessidades do processo actual:

- Maior acessibilidade das solicitações;
- Desburocratização e optimização do processo;
- Rapidez na busca de determinadas informações;
- Menor tempo de resposta para conclusão das solicitações;
- Maior segurança dos dados;
- Maior agilidade no processo;
- Restrições de acesso aos dados, fornecendo maior segurança;
- Eliminações de suporte papel,

3.2.4. Solução Proposta

O sistema desenvolvido nesse trabalho visa a informatização da documentação (contrato) no Departamento de Recursos Humanos.

De acordo com o contexto apresentado a solução proposta para o desenvolvimento do novo sistema apresenta as seguintes características

3.2.5. Análise do Sistema

As actividades de análise e projecto ajudam a transformar os requisitos do sistema em um projecto que pode ser implementado em software (CONNALEN, 2003). Análise e projecto podem ser feitos separadamente ou combinados como parte de um conjunto de actividades. A análise do caso de uso compreende aquelas actividades que levam os casos de uso e requisitos funcionais a produzirem um modelo de análise do sistema.

Nesta fase foi possível identificar e documentar módulos do sistema e a forma de interacção com o utilizador. Na modelagem dos diagramas, utilizando a notação UML (CONALLEN, 2003), foi possível determinar o comportamento geral do sistema, facilitando assim a fase de projecto. Durante a análise do sistema foram obtidos os diagramas de caso de uso gerais, os diagramas de classes e os modelos relacionais de bases de dados

3.2.6. Projecto Conceptual

A figura 4 mostra a solução proposta através do diagrama de casos de uso, onde o mesmo determina quais são os actores envolvidos no sistema (pessoas que interagem com o sistema), os módulos que tem acesso e suas funcionalidades.

Diagrama de Casos de Utilização

Os Casos de Uso têm por finalidade especificar o comportamento do sistema, sendo uma descrição de um conjunto de acções. Ele especifica o comportamento do sistema, sem ser necessário descrever como esse comportamento será implementado. A vantagem disso é a possibilidade do desenvolvedor se comunicar com o cliente sem se preocupar com os detalhes de como será implementado determinado comportamento do sistema (BOOCH, 2000).

A aplicação apresenta três tipos de utilizadores que podem ter acesso ao sistema; o funcionário que regista e gere contrato dos funcionários, administrador do módulo que pode também registar novos funcionários e consultar os registos e excluir, e o administrador do sistema que tem acesso a todos os módulos e a todas as funcionalidades.

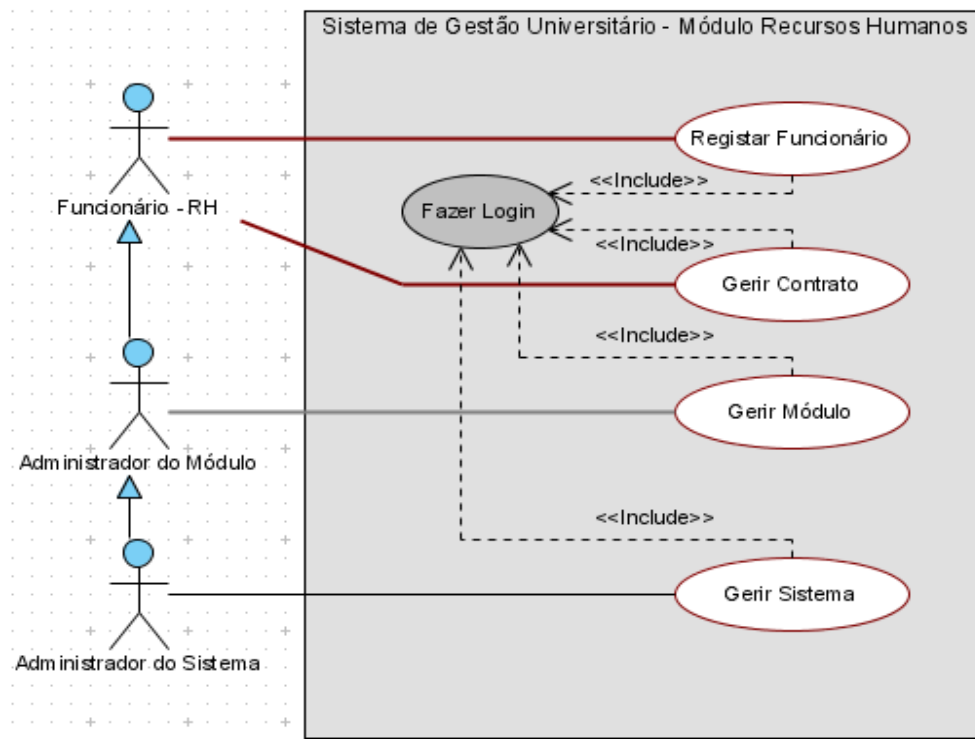



Figura 4: Diagrama de Casos de Utilização do Módulo Recursos Humanos

3.2.7. Descrição de Casos de Utilização

Caso de utilização – Login

Fluxo de Evento Principal

1. O utilizador introduz o nome do utilizador e senha;
2. Verifica-se na base de dados a existência do nome utilizador e senha
3. Direcciona para sua respectiva interface
4. Fim do caso de uso.

Fluxo de Evento Secundário

-Caso senha incorrecto

Acontece depois de 2

1.Envia msg de erro na mesma página;

Caso de utilização – Controlar Utilizador:

1. O administrador selecciona a opção de registo
2. Preenche o formulário do registo
3. Selecciona tipo de utilizador
4. Cria um novo login para o utilizador
5. O administrador selecciona a opção de alterar o registo
6. Procura o registo a ser alterado
7. Realiza a alteração
8. O administrador selecciona a opção de excluir registo
9. Procura o registo a ser excluído
10. Exclui o registo
11. Fim do caso de uso

Diagrama de Sequencia

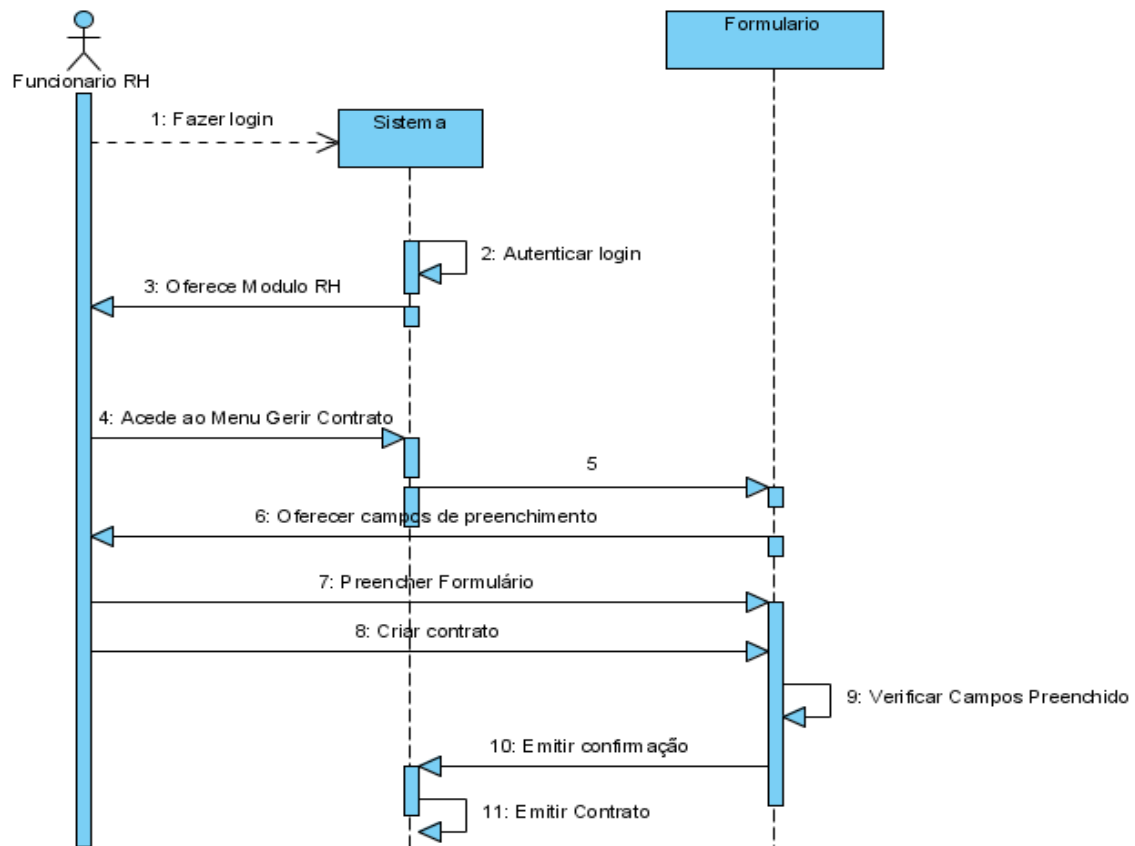



Figura 5: Diagrama de Sequencia do Módulo Recursos Humanos

Diagrama de Actividade

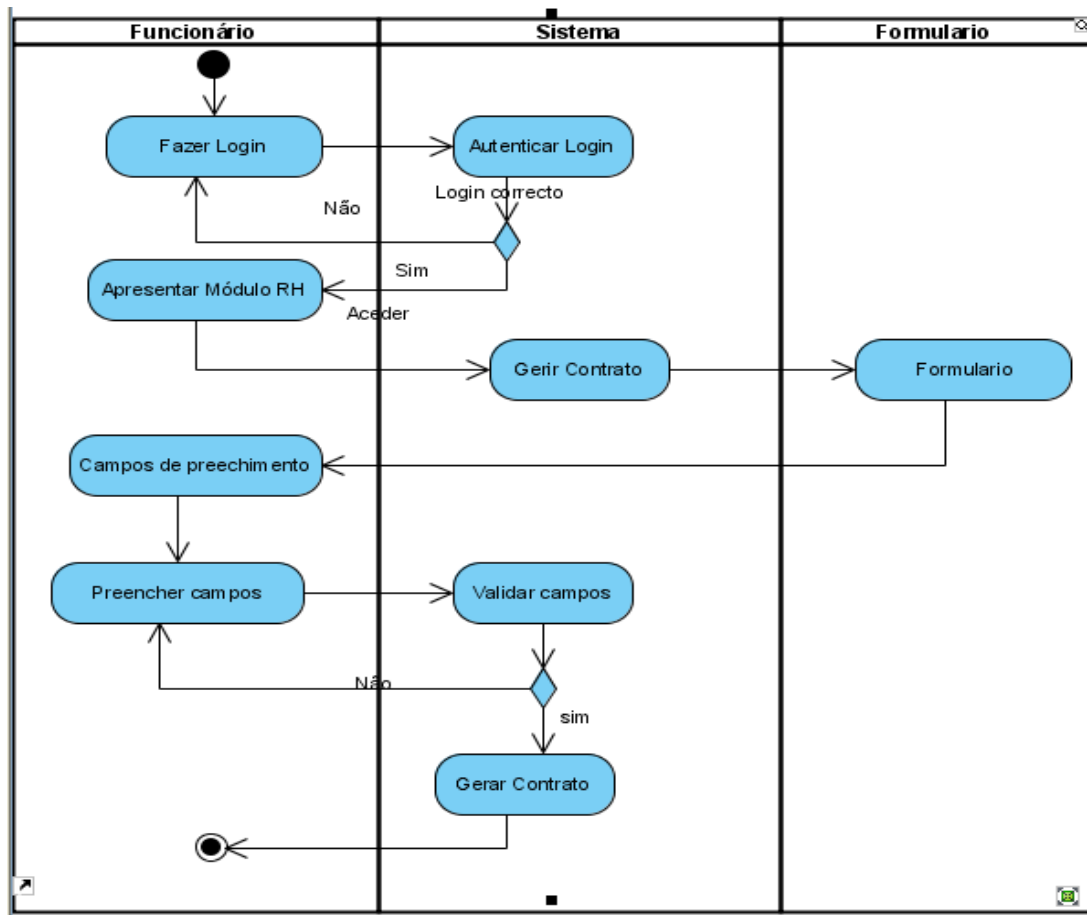



Figura 6: Diagrama de Sequencia do Módulo Recursos Humanos

Projecto Lógico

A figura 7 mostra as tabelas que compõem a base de dados, seus atributos e os respectivos relacionamentos (Diagrama Entidade-Relacionamento). (CHEN, 90)

Segundo SILBERCHATZ (2001), o modelo entidade relacionamento (E-R) tem por base a percepção de que o mundo real é formado por um conjunto de objectos chamadas entidades e pelo conjunto dos relacionamentos entre esses objectos. Descreve a comunicação e as relações entre os dados de um sistema. Os componentes que formam o diagrama são as entidades, relacionamentos e os atributos

Sua aplicação é para visualizar o relacionamento entre tabelas de uma base de dados, na qual as relações são construídas através da associação de um ou mais atributos destas tabelas.

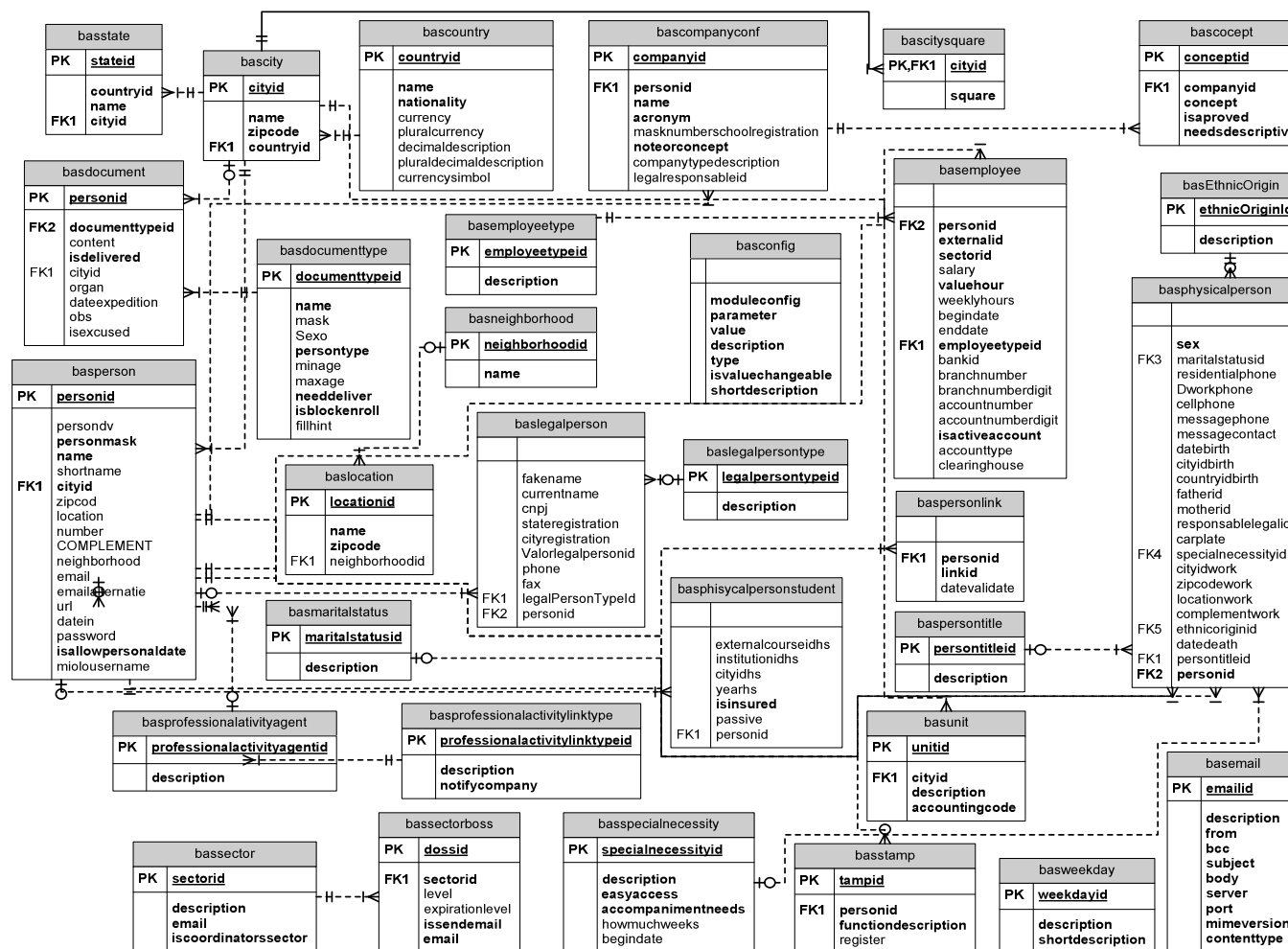



Figura 7:Modelo E-R

Para identificar os relacionamentos entre tabelas, são utilizadas chaves. Elas são campos que têm como característica identificar as tabelas de bases de dados através das chaves primárias e as relações entre tabelas através das chaves estrangeiras.

As chaves primárias são aquelas que identificam as tabelas em um determinado bases de dados. As chaves estrangeiras são utilizadas para visualizar relacionamentos entre tabelas, onde as chaves primárias de uma tabela são atributos de outra tabela. Essas chaves estrangeiras não necessariamente devem ser chaves primárias nas entidades que as recebem, mas devem ser na entidade que as originam.

3.2.8. Requisitos de Software e de Hardware necessário para a instalação do SGU

Requisitos de software:

- Servidor *web Apache* 2.2
- Interpretador PHP 5.2.4
- Banco de dados *PostgreSQL* 8.2
- Ágata Report
- *Ubuntu* 8.04

Requisitos de hardware:

Configuração mínima para funcionamento do SAGU:

- Processador Pentium IV ou similar
- RAM de 512MB (1GB recomendado)
- HD de 80GB (SATA recomendado)

Esta é a configuração mínima, geralmente utilizada apenas em ambiente de testes. Para utilização em ambiente de produção, recomenda-se um servidor de maior capacidade de processamento e armazenamento. Tal servidor deve ser dimensionado de acordo com a quantidade prevista de utilizadores que terão acesso simultâneo ao sistema.

Todos os computadores já estavam no local do estágio

Para a configuração do nosso trabalho foi utilizado:

- Processador Intel Pentium ® dual core CPU 180 GHz
- Memória RAM 1 GB
- Disco Rígido 160 GB

3.2.9. Requisitos Funcionais Do Sistema

- **Tipos de requisitos**

Os requisitos são geralmente classificados de acordo com a sua especificidade. Os tipos de requisitos mais gerais e vastamente utilizados são (JACOBSON, 2099):

Requisitos funcionais: especificam uma acção que o sistema deve ser capaz de fazer, sem quaisquer restrições físicas, tais como comportamentos de entrada e saída do sistema, lógica de decisão ou algoritmos.

Requisitos não-funcionais: especificam propriedades do sistema, tais como restrições de implementação, desempenho, segurança, manutenção ou extensão e que especificam restrições físicas aos requisitos funcionais

Requisitos funcionais do sistema:

- O sistema deve permitir o upload de arquivos;
- O sistema deve permitir a modificação de dados dos documentos;
- O sistema deve dar condições de aprovação ou rejeição de documentos pelos utilizadores;
- O sistema deve gerir contratos;
- O sistema deve manter um histórico dos eventos;
- O sistema deve prover acesso a documentos pelos utilizadores de acordo com seu nível de privilégio e departamentos;
- O sistema deve permitir o registo de utilizadores;
- O sistema deve controlar o acesso dos utilizadores através de login e senha;
- O sistema deve permitir que o utilizador altera sua senha;
- O sistema deve permitir a modificação de dados do utilizador;
- O sistema deve permitir o registo dos *backups*.

Requisitos não-funcionais do sistema:

- O sistema deverá operar em plataforma GNU/LINUX;

3.2.10. Actividades Planeadas

Cronograma

A seguir é apresentado o cronograma das actividades planeadas, iniciando em Julho de 2009 e terminando em Dezembro de 2009, com a duração total do trabalho de 265 Horas.

Nome da Tarefa	Duração	Início	Conclusão	Pred
Implementação do Sistema de Gestão Universitário	**97 dias**	**03-08-09**	**15-12-09**	
Recolha de Informação	**43 dias**	**03-08-09**	**30-09-09**	
Pesquisas Bibliográficas	43 dias	03-08-09	30-09-09	
Leitura da documentação	15 dias	05-08-09	25-08-09	
Conhecer as funcionalidades do Sistema	15 dias	05-08-09	25-08-09	
Customização e Implementação do Módulo Recursos Humanos	**64 dias**	**02-09-09**	**30-11-09**	
Levantamento e análise de requisitos	3 dias	02-09-09	04-09-09	5
Instalação de Sistema	**5 dias**	**07-09-09**	**11-09-09**	
Instalação	2 dias	07-09-09	08-09-09	7
Tradução do Sistema	2 dias	09-09-09	10-09-09	9
Colocação do Sistema na Rede	1 dia	11-09-09	11-09-09	10
Modelação do Módulo	**25 dias**	**14-09-09**	**16-10-09**	
Concepção do diagrama Modelo E-R	7 dias	14-09-09	22-09-09	11
Diagarmas de Casos de Usos	5 dias	23-09-09	29-09-09	13
Diagramas de actividades	3 dias	30-09-09	02-10-09	14
Diagramas de sequência	2 dias	05-10-09	06-10-09	15
Implementação	**8 dias**	**07-10-09**	**16-10-09**	
Equiparação e Geração de tabelas relacionais do SAGU com SGU	5 dias	07-10-09	13-10-09	16
Customização de Layout	3 dias	14-10-09	16-10-09	18
Testes e correcções de erros	**39 dias**	**07-10-09**	**30-11-09**	
Testes	39 dias	07-10-09	30-11-09	
Correção de erros	39 dias	07-10-09	30-11-09	
Redação e correção de relatório final	69 dias	10-09-09	15-12-09	

Figura 8: Actividades Planeadas

Diagrama da Gantt referente à calendarização das tarefas descritas no estagio

3.3. Actividades Desenvolvidas

Numa primeira fase, que teve duração de duas semanas, foram consolidados os conhecimentos do sistema operativo Linux ao nível da utilização, assim como aspectos fundamentais, quer do hardware, quer do software, inerentes ao sistema.

Posteriormente, e durante um meses, no âmbito da administração Linux, foram desenvolvidas as seguintes tarefas:

- Como o sistema deveria ser baseado na plataforma Web, foi feita uma análise do sistema em questão

- instalação de software, onde houve dificuldades, pois foi primeira vez que tive oportunidade de instalar software que depende de outros pacotes. Isso foi superado com instruções do co-orientador;

- gestão de utilizadores e de grupos, gestão de espaço em disco, execução de cópias de segurança ("*backups*"), e correcções de problemas pontuais.

- Desenvolvimento da documentação do projecto através de diagramas UML. Os diagramas desenvolvidos foram o de Classes que facilita a identificação de associações e atributos e os Diagramas de Actividades e de Casos de Utilização que juntamente com o de Classe auxiliam na compreensão do funcionamento do sistema. Estes diagramas foram desenvolvidos por outros integrantes do projecto. Além dos diagramas UML, também foi desenvolvido o DER que, juntamente com o Diagrama de Classes são responsáveis por mostrar uma visão da estrutura do banco de dados.

- geração contrato escrito através do PHP, também houve dificuldades; foi superado com pesquisas na internet, apoios do co-orientador e colegas.

- a implementação do projecto, onde foi seguida a metodologia UML

- desenvolvimento da camada visual do sistema. (interfaces)

3.4. Dificuldades Encontradas

Uma das primeiras dificuldades encontradas para o desenvolvimento deste trabalho com certeza foi a falta de experiência para o desenvolvimento de um sistema desse porte. Mas isso foi superado aos poucos, graças às pesquisas realizadas e às trocas de experiências com a equipa de implementação do sistema.

Falta de documentação também foi um dos problemas encontrados.

3.5. Implementação

A linguagem de programação escolhida para o desenvolvimento foi o PHP (WELLING, 2003), por se tratar de uma linguagem ideal para esse tipo de sistema e ainda ser de uso livre. Apesar de não ser uma linguagem totalmente orientada a objectos, a versão 5 do PHP (WELLING, 2003) oferece vários recursos que permitem uma implementação muito próxima da orientação a objectos, como a utilização de classes e encapsulamento, por exemplo.

A base de dados utilizados foram o *PostgreSQL* (GROUP, 2002), para a intranet, por ser uma base de dados de licença livre que mais oferece recursos, (WELLING, 2003), para a Home Page, por ser o único disponível no host onde está hospedada. O sistema interno executa em um servidor Http com sistema operativo GNU/Linux, (*Ubuntu*) e é acessado pelos utilizadores através de um navegador Web

3.5.1. Interfaces do Sistema Proposto

A figura 9 mostra a interface de autenticação.

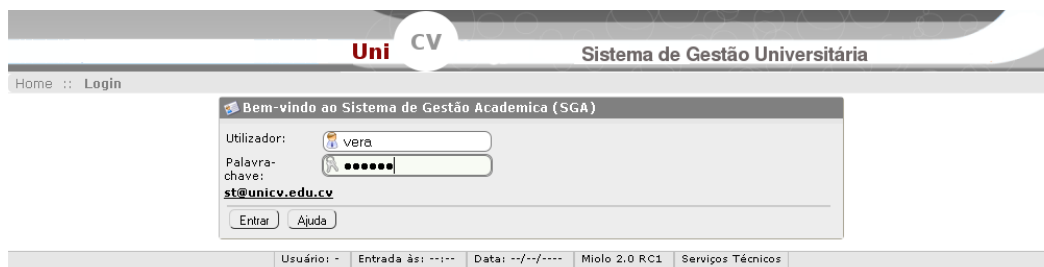



Figura 9: Validação de login

Fonte: SGU

Figura 10: Janela Principal

Fonte: SGU

Figura 11: Janela de Recursos Humanos

Fonte: SGU

3.5.2. Testes

Não foi utilizado nenhum método específico para realização dos testes. Todos os testes foram realizados utilizando registos reais de documentos no sistema, contando com a colaboração do RH para inserção e registos de todos documentos.

Com a execução dos testes foi possível descobrir erros referentes, basicamente, à programação, conexões e consultas a bases de dados e desempenho do sistema no processamento e apresentação das informações. Todas as falhas encontradas durante a

programação e testes foram corrigidas, obtendo-se um funcionamento bastante satisfatório do sistema.

3.5.3. Redacção do Trabalho

A redacção do trabalho foi feita paralelamente a todas fases do projecto, iniciando durante a fase de análise do sistema, e finalizado depois do *software* testado, implantado e de se ter certeza sobre o seu bom funcionamento.

O trabalho final contém toda documentação gerada nas fases de Análise do Sistema, Projecto e Análise dos Resultados, bem como registos das dificuldades encontradas e conclusões sobre o desenvolvimento. A elaboração do trabalho foi realizada utilizando norma técnicas fornecida através de exemplos de trabalhos com metodologia científica.

IV. CONCLUSÃO

4.1. Conclusão

Neste capítulo, são apresentadas as conclusões deste trabalho e recomendações de trabalhos futuros.

4.2. Considerações gerais

Fazer um estudo e desenvolver/implementar um sistema de gestão na Universidade, como o processo de comunicação institucional, é uma tarefa no mínimo desafiadora. Ao se propor ainda, a utilização dos conceitos de TI e SI, que são áreas de constante actualização, o desafio torna-se promissor.

Várias são as tendências e abordagens de sistema de gestão, mas todas tratam este processo de maneira essencial para a manutenção de um empreendimento. É inevitável, ao tratar deste assunto, deparar-se com outros dois termos totalmente relacionados: informação e conhecimento. E este trabalho alcançou seus objectivos ao cumprir o prazo e deixar o sistema em funcionamento.

Como contribuição, foi obtido o Sistema de Gestão Universitária (SGU), com utilização na Universidade. Sendo a descrição das ferramentas utilizadas no desenvolvimento e no próprio estudo de caso apresentadas neste trabalho.

O SGU forneceu à Universidade meios de fortalecimento do processo comunicacional, apoio na disseminação do conhecimento e tomada de decisões, partilha da informação e consequente aumento de produtividade dos colaboradores.

Estes por sua vez, beneficiaram-se com a obtenção de um ambiente único detentor de grande parte do conteúdo utilizado no trabalho. A área de tecnologia da Universidade beneficiou-se com a confiança gerada pela estabilidade da ferramenta e pela experiência adquirida ao longo de todas as fases de desenvolvimento e/ou implementação.

Durante o trabalho observou-se que a informação é um recurso muito valioso e portanto deve ser explorada pelas empresas para que forneça um valor a seus produtos ou serviços.

Actualmente, esta tecnologia tem uma importante função nas Universidades possibilitando rapidez no relacionamento entre a Universidade e o Cliente.

Com o desenvolvimento do sistema, explanado neste trabalho, foi possível a informatização da documentação.

4.3. Trabalho Futuro

A implementação de outras tecnologias e funcionalidades no sistema apresentado, neste trabalho, podem contribuir para o seu enriquecimento e utilidade, gerando diferentes resultados e alcançando um vasto leque de objectivos.

Entre as recomendações para trabalhos futuros tomando como base a proposta abordada, incluem-se:

- Disponibilização de acesso ao conteúdo via Internet, ou seja, de qualquer lugar o utilizador pode aceder ao ambiente da Intranet da Universidade;
- A utilização de conceitos e técnicas de *Data Mining*, que consiste na mineração de dados que fornecem base para tomadas de decisões;
- Modelar o sistema de modo que o sistema faz a contagem de tempo de serviço e produtividade dos funcionários.
- Criar um manual de ajuda para os utilizadores do módulo Recursos Humanos.

4.4. Apreciação final

Acerca do projecto pode-se dizer que se tratou de um trabalho que alargou a nossa experiência ao nível de desenvolvimento de sistemas. Pôs em prática grande parte dos conhecimentos de certas disciplinas do curso, explorando-os ainda mais.

Para além dessa evolução, evidente, a nível pessoal não se pode deixar de referir o facto de contribuir bastante para a instituição, pois era esse o seu fim. Tratou-se de um trabalho teórico e prático, o que o tornou mais interessante. À medida que as parametrizações eram feitas, o resultado era imediatamente visível.

É necessário citar que o desenvolvimento deste trabalho possibilitou a aplicação prática de muitos conceitos aprendidos em sala de aula, nas mais diversas disciplinas, e ainda

possibilitou a agregação de conhecimentos extras, devido às dificuldades encontradas e às pesquisas necessárias para superá-las.

BIBLIOGRAFIA

ADVOCACY. As vantagens do PostgreSQL 2002. Disponível em: http:// advocacy. Postgresql.org/advantages/?lang=br. Consultado em: 29 Agosto 2009.

AFONSO. M. Gestão escolar: revendo conceitos. São Paulo, PUC-SP, 2004. Disponível em: http://www.gestores.pucsp.br/. Consultado em: 10 de Agosto de 2009.

ANYSOFT. Informática, 2005 Disponível em http:// www. anysoft . com . br/home/content/view/13/52/, Consultado em 18 de Agosto de 2009

APACHE. "The Apache Software Foundation". Disponível em: <http://www.apache.org/>. Consultado em: 09 de Outubro de 2009.

APOSTILA. UBUNTU – LINUX , Setembro 2006, Disponível em: http://guaiba. ulbra.tche. br/ danielep/ APOSTILA % 20 LINUX.pdf, Consultado em 31de Agosto de 2009

APOSTILA. UBUNTU – LINUX Disponível em: http://www.ubuntubrasil.org/ubuntu, Consultado em 18 de Agosto de 2009

BOOCH, G. et al. UML: Guia do usuário. Rio de Janeiro: Campus, 2000, 215 p.

BOOCH. Grady; RUMBAUGH, James; JACOBSON, Ivar. **UML:** Guia do Utilizador. Tradução de Fábio Freitas. 11 tiragem. Rio de Janeiro: Campus, 2000.

BROD. César, Revista do Linux, Disponível em: http://augustocampos.net/revista-do-linux/ 035/ software .html, Consultado em 26 de Agosto de 2009

CANONICAL. Ltd.Curso Básico do Ubuntu 8.04 Disponível en http://andregondim.eti.br/?p=384. Consultado em 26 de Agosto de 2009

DAMASIO. E.; RIBEIRO, C. E. N. Software livre para bibliotecas, sua importância e utilização: o caso Gnuteca. In: XXI Congresso Brasileiro de Biblioteconomia, Documentação e Ciência da Informação, 2005, Disponível em: <http://br.geocities.com/edilson_ amasio/

NUTEC.pdf>. Consultado em 21 de Outubro 2009 em**:** http://www. guiadopc.com. br/download / Manual %20Ubuntu%208.04.pdf, Consultado em Agosto 2009

FERNANDES. Anilton S. 2008, Tecnologias Livres de Desenvolvimento Web. Disponível em: http:// bdigital. unipiaget.cv:8080/dspace/bitstream/.pdf, Consultado em 14 de Setembro de 2009

GNU. (2004). GNU General Public License. (online), disponível em http://www. gnu. org/ opyleft/gpl.html consultado em 2009

GNUTECA. Sistema de Gestão de Acervo, Empréstimo e Colaboração para Bibliotecas. Disponível em: http://www.gnuteca.org.br. Consultado em: 20 Outubro de 2009

HESSEL. A. As TIC podem auxiliar na gestão da escola. São Paulo: PUC-SP, 2004. Disponível em: http://www.gestores.pucsp.br/. Acesso em: 30 Julho de 2009

KRUCHTEN, P. The Rational Unified Process: an introduction. Boston: Addison-Wesley, 1999.

KRUCHTEN, Phillippe. **Introdução ao RUP - Rational Unified Process.** 2. ed. Nacional: Ciencia Moderna, 2003. 272 p.

KARADIMA. 1992, O. Administracion y planificacion universitaria: El principia del liderazgo académico. Santiago de Chile: Seminário – Planificación Estratégica Universitária, 1992. Disponível em: www. abepro. org. br/biblioteca /ENEGEP 2001_ TR72_0180.pdf. Consultado em: 30 Julho de 2009

KINOSHITA. V.G. “Banco de Dados Via Web: uma Análise Comparativa”. Universidade Federal de Lavras, 2001.

LÉVY. P. As Tecnologias da Inteligência: o Futuro do Pensamento na Era da Informática São Paulo: Editora 34, 1996.

LUZ. Fernanda N. Sistema de Controlo de Apoio à Escola Utilizando Software Livre http://biblioteca.claretiano.edu.br/phl8/pdf/20001964.pdf, Consultado em 19 de Novembro de 2009

PEREIRA. F. (2003). *Linux - Curso Completo, 4ª Edição Actualizada*, Editora FCA, ISBN

PGADMIN III. Disponível em: http://www.pgadmin.org/pgadmin3/index.php. Consultado em: 29 de Agosto de 2009.

PHP. "PHP: Hipertext Preprocessor". Disponível em: <http://www.php.net/>. Consultado em: 10 de Setembro de 2009.

PHP. (2004). *Hypertext Preprocessor*. (online). Disponível em http://www.php.net/ Consultado em 24 de Agosto de 2009

POSTGRESQL GLOBAL DEVELOPMENT GROUP. "Tutorial do PostgreSQL 8 /Official Documentation". Tradução: Halley Pacheco de Oliveira. Rio de Janeiro, 2005. Disponível em: http://www.postgresql.org/. Consultado em 10 de Setembro de 2009.

POSTGRESQL. Disponível em:http://www.postgresql.org/. Consultado em 29 de Agosto de 2009

POSTGRESQL. site da documentação oficial (http://www.postgresql.org/docs/)

QUATRANI, T. Visual modeling with Rational Rose and UML. Boston: Addison-Wesley, 1999.

REIS. Christian Robottom . "Caracterização de um Processo de Software para Projectos de Software Livre". Dissertação de mestrado. USP, São Paulo, 2003.

RATIONAL SOFTWARE. Rational Unified Process. IBM, 2005. Acessado em 27/08/2009. Disponível em http://www.ibm.com/rational/rup.

SAGU (2007) "Sistema Aberto de Gestão Unificada".Disponível em http://sagu.solis.coop.br/. Consultado em 22 de Julho de 2009

SILVA. Patrícia S. M. Nunes da, Análise de Soluções Open Source para Administração de Redes 2006. Disponível em: http://mosel.estg.ipleiria.pt/files/Artigo_Sysadmin.pdf, Consultado em 17 de Novembro de 2009

SOLIS – Cooperativa de Soluções Livres, SAGU - Sistema Aberto de Gestão Unificada, Manual para Usuários, Consultado em 24 de Julho de 2009

SOLIS 2009. Disponível em http://sagu2.solis.coop.br/, Consultado em 15 de Julho de 2009

SOUZA, S. A. F. Gestão educacional na era da informação: novas competências e competências reconfiguradas. Disponível em: http://www.elton.com.br/FSDB/TI- FSDB-Esp-Texto5-Souza.pdf. Consultado em: 10 de Agosto de 2009.

SOUZA, Thiago Silva, Rational Unified Process: uma abordagem gerencial http://www. cesarkallas.net/arquivos/faculdade/tcc/teoria/METODOLOGIAS.doc Consultado em 23 de Julho de 2009

TEIXEIRA, Agostinho 2006, Portal De Tutoriais Open Source. Disponível em: http://mosel. estg.ipleiria. pt/ files / relatorio_tutoriais.pdf, Consultado em 15 de Setembro de 2009

Universidade Estadual De Goiás projecto Para Desenvolvimento E Implantação Do Sistema De Gerenciamento De Actividades Docentes. Disponível em http:// www.radoc.ueg. br/_ arquivos/ sistema_ sigad.pdf; Consultado em 23 de Julho de 2009.

Universidade Estadual De Goiás projecto Sistemas Integrados. Disponível em http:// www. Radoc. ueg.br/ _arquivos/ 2008/resumo_ projecto_sigad.pdf, Consultado em 23 de Julho de 2009

VIEIRA, A. Funções e Papéis da Tecnologia. São Paulo, PUC-SP, 2004. Disponível em: http://www.gestores.pucsp.br/. Consultado em: 10 de Agosto de 2009.

ANEXO

Anexo A - Tabelas do Modulo Básico – utilizado no Modulo Recursos Humanos

Tabela nº 1

Basaccess/tabela que guarda os links que os utilizadores acederam			
Campo	**Tipo**	**Chave**	**Descrição**
login	text		Nome do utilizador que efectuou o acesso
moduleaccess	text		Módulo que o utilizador acessou
label:	text		Rotulo
image	text		Imagem
handler:	text		Handler acessado
isbookmark	boolean		Se o registo e um registo dos favoritos do utilizador

Tabela nº 2

Bascity/cidades			
Campo	**Tipo**	**Chave**	**Descrição**
cityid	integer	PK	Código da cidade
name	text		Nome
zipcode:	Char(9)		CEP
stateid:	Char(2)		Código do estado/província da federação
countryid:	integer	FK	Código do país

Tabela nº 3

Bascitysquare/praças das cidades para o banco			
Campo	**Tipo**	**Chave**	**Descrição**
cityid	integer	PK, FK	Código da cidade
square:	Char(6)		Praça

Tabela nº 4

Bascompanyconf/Registo da empresa que utiliza o sistema.			
Campo	**Tipo**	**Chave**	**Descrição**
companyid	integer	PK	Código da empresa
personid	integer	FK	Código da pessoa para buscar os dados da pessoa jurídica da Instituição
name	Text		Nome da empresa
acronym:	Char(10)		Sigla da empresa
masknumberschoolregistration	Text		Mascara para o número de registo escolar
noteorconcept	Char(1)		Se utiliza nota ou conceito
companytypedescription	Text		Descrição do tipo de instituição. Ex: Centro universitário, universidade, Faculdade
legalresponsableid:	integer		Responsável legal pela instituição

Tabela nº 5

Basconcept/conceitos utilizados pela instituição			
Campo	**Tipo**	**Chave**	**Descrição**
conceptid:	integer	PK	Código do Conceito
companyid	integer	PK,FK	Código da empresa
concept	Char(15)		Conceito
isapproved:	Boolean		Se e um conceito de aprovação
needsdescriptive	boolean		E necessário descritivo

Tabela nº 6

Basconfig/configurações do sistema			
Campo	**Tipo**	**Chave**	**Descrição**
moduleconfig	text	PK	Módulo do parâmetro
parameter	text	PK	Parâmetro
value:	text		Valor
description	text		Descrição do parâmetro
type:	Char(50)		Tipo do parâmetro

Tabela nº 7

Bascountry/países			
Campo	**Tipo**	**Chave**	**Descrição**
countryid	integer	PK	Código do país
name	Char(20)		Nome
nationality:	Char(20)		Nacionalidade
currency	Char(20)		Moeda no singular. Ex: ECV
pluralcurrency	Char(20)		Moeda no plural. Ex: ECV
decimaldescription	Char(20)		Como são chamados os valores decimais no singular. Ex: CENTAVO
pluraldecimaldescription:	Char(20)		Como são chamados os valores decimais no plural. Ex: CENTAVOS
currencysymbol:	Char(20)		Símbolo da moeda. Ex: ECV

Tabela nº 8

Bascvslog/tabela de log de alterações na base através dos processos on-line			
Campo	**Tipo**	**Chave**	**Descrição**
cvslogid	integer	PK	Código do log
tablename	text		Nome da tabela alterada
fieldname	text		Nome do campo alterado
oldvalue:	text		Valor antigo
tablepkey	text		Chave primária da tabela
tablepkeyvalue	text		Valor da chave primária da tabela

Tabela nº 9

Basdocument/documentos dos alunos			
Campo	**Tipo**	**Chave**	**Descrição**
personid:	integer	PK,FK	Código da pessoa
documenttypeid	integer	PK,FK	Código do tipo do documento
content	text		Numero, texto ou valor do documento
isdelivered:	boolean		Se o documento foi entregue ou não
cityid	integer	FK	Cidade de expedição do documento, se for necessário
organ:	Char(15)		Órgão Expedidor do documento, se for necessário
dateexpedition:	date		Data de expedição do documento, se for necessário
obs:	text		Eventual observação
isexcused:	boolean		Campo para setar se determinada pessoa está dispensada de apresentar este documento

Tabela nº 10

Basdocumenttype/registo dos tipos de documento			
Campo	**Tipo**	**Chave**	**Descrição**
documenttypeid	integer	PK	Código do tipo de documento
name	Text		Nome do documento
mask:	Text		Mascara de entrada
sex:	Char(1)		Sexo da pessoa (M, F ou All)
persontype:	Char(1)		Tipo da pessoa (física (P), jurídica (L) ou todas (All))
minage	integer		Idade mínima para preenchimento deste campo
maxage	integer		Idade máxima para preenchimento deste campo
needdeliver	Boolean		Se vai ser necessário entregar uma cópia do mesmo ou não
isblockenroll	boolean		Documentos que bloqueiam o processo de matricula
fillhint:	text		Texto auxiliar exibido como dica para preenchimento do documento. Pode Ser, por exemplo, a máscara de preenchimento

Tabela nº 11

Basemail/tabela que grava o template dos emails a serem enviados e suas configurações			
Campo	**Tipo**	**Chave**	**Descrição**
emailid:	integer	PK	Código do email
description	text		Descrição do que e este email
from	text		Remetente
bcc:	text		Cópia oculta
subject	text		Assunto do email
body:	text		Corpo do email
server:	text		Servidor de envio do email
port:	integer		Porta de envio do email no servidor
mimeversion:	text		MIME-version do email
contenttype	text		Content-Type do email

Tabela nº 12

Basemployee/pessoa física - funcionário			
Campo	**Tipo**	**Chave**	**Descrição**
employeeid	integer	PK	Código do funcionário
personid:	integer	FK	Código da pessoa
externalid:	Char(10)		Código no Siga
sectorid:	integer	FK	Código do sector
salary:	Numeric(14,2)		Salário
valuehour:	boolean		Se o salário equivale ao valor da hora?
weeklyhours	Double precision		Quantas horas semanais
begindate:	Date		Data inicial deste vínculo
enddate	date		Data final deste vínculo
employeetypeid:	integer	FK	Tipo de funcionário. Ex: Estagiário, funcionário, bolsista
bankid	Char(3)	FK	Banco da conta do funcionário
branchnumber	Char(20)		Número da agência
branchnumberdigit	Char(2)		Dígito da agência
accountnumber:	Char(20)		Número da conta
accountnumberdigit	Char(2)		Dígito da conta
isactiveaccount:	boolean		Se a conta esta activa ou não
accounttype:	Char(2)		Tipo da conta - 01: Conta Corrente 05: Conta Poupança
clearinghouse:	integer		Camera de Compensação: Informação utilizada no arquivo de abertura de contas

Tabela nº 13

Basemployeetype/Tipo de funcionário			
Campo	**Tipo**	**Chave**	**Descrição**
employeetypeid	integer	PK	Tipo de funcionário
description	text		Descrição

Tabela nº 14

Basethnicorigin/Origem étnica			
Campo	**Tipo**	**Chave**	**Descrição**
ethnicoriginid:	integer	PK	Código da origem étnica
description	text		Descrição

Tabela nº 15

Basfile/Tabela que grava os arquivos de upload			
Campo	**Tipo**	**Chave**	**Descrição**
fileid	integer	PK	Código do arquivo
filename	text		Nome do arquivo

Tabela nº 16

Baslegalperson/pessoas jurídicas			
Campo	**Tipo**	**Chave**	**Descrição**
personid	integer	PK	Código da pessoa
persondv	integer		Dígito verificador da pessoa
personmask	Char(15)		Mascara para o Código da pessoa
name	Char(100)		Nome
shortname	Char(30)		Apelido
cityid	integer		Código da cidade
zipcode	Char(9)		CEP
location	Char(100)		Logradouro
number	Char(50)		Numero
complement	Char(60)		Complemento
neighborhood	text		Bairro
email	Char(60)		Email
emailalternative	Char(60)		Email alternativo
url	Char(60)		URL
datein	date		Data de ingresso
password	Char(10)		Senha para acesso aos processo on-line
isallowpersonaldata	boolean		Permite a divulgação de dados pessoais
miolousername	Char(25)		Nome do utilizador que eh inserido automaticamente na base de dados do Miolo para acesso ao módulo de serviços
fakename	Char(100)		Nome fantasia
currentname	Char(100)		Nome actual
cnpj	Char(20)		CNPJ
stateregistration	Char(20)		Inscrição estadual
cityregistration:	Char(20)		Inscrição municipal
legalpersontypeid	integer	FK	Tipo de pessoa jurídica
phone:	Char(50)		Telefone
fax:	Char(50)		Fax

Tabela nº 17

Baslegalpersontype/tipos de pessoa jurídica			
Campo	**Tipo**	**Chave**	**Descrição**
legalpersontypeid	integer	PK	Código do tipo de pessoa jurídica
description	text		Descrição

Tabela nº 18

Baslink/vínculos			
Campo	**Tipo**	**Chave**	**Descrição**
linkid	integer	PK	Código do vínculo
description	text		Descrição

Tabela nº 19

Baslocation/concelho			
Campo	**Tipo**	**Chave**	**Descrição**
locationid:	integer	PK	Código do logradouro
name:	text		Nome
cityid:	integer	FK	Código da cidade
zipcode	Char(9)		CEP
neighborhoodid	integer	FK	Código do bairro

Tabela nº 20

Baslog/*registos de utilizadores e data de alteração*			
Campo	**Tipo**	**Chave**	**Descrição**

Tabela nº 21

Basmaritalstatus/estado civil			
Campo	**Tipo**	**Chave**	**Descrição**
maritalstatusid:	Char(1)	PK	Código do estado civil
description	text		Descrição

Tabela nº 22

Basneighborhood/bairros			
Campo	**Tipo**	**Chave**	**Descrição**
neighborhoodid:	integer	PK	Código do bairro
name:	text		Nome

Tabela nº 23

Basperson/pessoas			
Campo	**Tipo**	**Chave**	**Descrição**
personid	integer	PK	Código da pessoa
persondv:	integer		Dígito verificador da pessoa
personmask	Char(15)		Mascara para o Código da pessoa
name	Char(100)		Nome
shortname	Char(30)		Apelido
cityid:	integer	FK	Código da cidade
zipcode	Char(9)		CEP
location:	Char(100)		Logradouro
number:	Char(10)		Numero
complement	Char(60)		Complemento
neighborhood:	text		Bairro
email:	Char(60)		Email
emailalternative	Char(60)		Email alternativo
url:	Char(60)		URL
datein	date		Data de ingresso
password:	Char(10)		Senha para acesso aos processo on-line
isallowpersonaldata:	boolean		Permite a divulgação de dados pessoais
miolousername	Char(25)		Nome do utilizador que eh inserido automaticamente na base de dados do miolo para acesso ao modulo de servicos

Tabela nº 24

Baspersonlink/vínculos das pessoas			
Campo	**Tipo**	**Chave**	**Descrição**
personid	integer	FK	Código da pessoa
linkid:	integer	FK	Código do vínculo
datevalidate	date		Data de validade

Tabela nº 25

Baspersontitle/Titulos das pessoas			
Campo	**Tipo**	**Chave**	**Descrição**
persontitleid	integer	PK	Código do título da pessoa
description	Char(15)		Descrição do título

Tabela nº 26

Basphysicalpersonemployee/pessoa física - funcionário			
Campo	**Tipo**	**Chave**	**Descrição**
personid	integer	PK	Código da pessoa
persondv	integer		Dígito verificador da pessoa
personmask	Char(15)		Mascara para o Código da pessoa
name	Char(100)		Nome
shortname:	Char(30)		Apelido
cityid	integer	FK	Código da cidade
zipcode:	Char(9)		CEP
location	Char(100)		Logradouro
number:	Char(10)		Numero
complement:	Char(60)		Complemento
neighborhood:	text		Bairro
email:	Char(60)		Email
emailalternative:	Char(60)		Email alternativo
url:	Char(60)		URL
datein	date		Data de ingresso
password:	Char(10)		Senha para acesso aos processo on-line
isallowpersonaldata:	boolean		Permite a divulgação de dados pessoais
miolousername	Char(25)		Nome do utilizador que é inserido automaticamente na base de dados do miolo para acesso ao modulo de serviços
sex:	Char(1)		Sexo
maritalstatusid:	Char(1)	FK	Estado civil
residentialphone:	Char(50)		Telefone residencial

workphone	Char(50)		Telefone comercial
cellphone:	Char(50)		Telefone celular
messagephone:	Char(50)		Telefone para recados
messagecontact:	Char(50)		Contacto para recados
datebirth	date		Data de nascimento
cityidbirth	integer	FK	Cidade de nascimento
countryidbirth	integer	FK	Pais de nascimento
fatherid:	integer	FK	Pai
motherid:	integer	FK	Mãe
responsablelegalid:	integer	FK	Responsável legal
carplate:	Char(40)		Placa do carro
specialnecessityid:	integer		Necessidade especial
specialnecessitydescription:	text		Descritivo da necessidade especial
cityidwork	integer		Cidade do local de trabalho
zipcodework:	Char(9)		CEP do local de trabalho
locationwork:	Char(100)		Logradouro do local de trabalho
complementwork:	Char(40)		Complemento do local de trabalho
neighborhoodwork	Char(100)		Bairro do local do trabalho
ethnicoriginid:	integer	FK	Origem étnica
datedeath	date		*Data de óbito do aluno*
persontitleid	integer	FK	*Título da pessoa*

Tabela nº 27

Basphysicalperson/pessoa física			
Campo	**Tipo**	**Chave**	**Descrição**
personid	integer	PK	Código da pessoa
persondv	integer		Digito verificador da pessoa
personmask	Char(15)		Mascara para o Código da pessoa
name	Char(100)		Nome
shortname	Char(30)		Apelido
cityid	integer		Código da cidade
zipcode:	Char(9)		CEP
location:	Char(100)		Logradouro
number:	Char(10)		Numero

complement:	Char(60)		Complemento
neighborhood:	text		Bairro
email:	Char(60)		Email
emailalternative:	Char(60)		Email alternativo
url:	Char(60)		URL
datein:	date		Data de ingresso
password:	Char(10)		Senha para acesso aos processo on-line
isallowpersonaldata:	boolean		Permite a divulgação de dados pessoais
miolousername	Char(25)		nome do utilizador que eh inserido automaticamente na base de dados do miolo para acesso ao modulo de serviços
sex:	Char(1)		Sexo
maritalstatusid	Char(1)	FK	Estado civil
residentialphone	Char(50)		Telefone residencial
workphone	Char(50)		Telefone comercial
cellphone:	Char(50)		Telefone celular
messagephone	Char(50)		Telefone para recados
messagecontact:	Char(50)		Contacto para recados
datebirth:	date		Data de nascimento
cityidbirth:	integer	FK	Cidade de nascimento
countryidbirth:	integer	FK	Pais de nascimento
fatherid:	integer	FK	Pai
motherid:	integer	FK	Mãe
responsablelegalid:	integer	FK	Responsável legal
carplate	Char(40)		Placa do carro
specialnecessityid:	integer		Necessidade especial
specialnecessitydescription	text		Descritivo da necessidade especial
cityidwork	integer		Cidade do local de trabalho
zipcodework:	Char(9)		CEP do local de trabalho
locationwork:	Char(100)		Logradouro do local de trabalho
complementwork:	Char(40)		Complemento do local de trabalho
neighborhoodwork:	Char(100)		Bairro do local do trabalho
ethnicoriginid	integer	FK	Origem étnica
datedeath:	date		Data de óbito do aluno

Persontitle	integer	FK	Título da pessoa

Tabela nº 28

Basphysicalpersonstudent/ pessoa física - estudante			
Campo	**Tipo**	**Chave**	**Descrição**
personid:	integer	PK	Código da pessoa
persondv	integer		Dígito verificador da pessoa
personmask:	Char(15)		Mascara para o Código da pessoa
name	Char(100)		Nome
shortname:	Char(30)		Apelido
cityid	integer		Código da cidade
zipcode	Char(9)		CEP
location:	Char(100)		Logradouro
number	Char(10)		Numero
complement	Char(60)		Complemento
neighborhood	text		Bairro
email	Char(60)		Email
emailalternative	Char(60)		Email alternativo
url:	Char(60)		URL
datein	date		Data de ingresso
password	Char(10)		Senha para acesso aos processo on-line
isallowpersonaldata	boolean		Permite a divulgação de dados pessoais
miolousername	Char(25)		Nome do usuário que eh inserido automaticamente na base de dados do miolo para acesso ao modulo de serviços
sex	Char(1)		Sexo
maritalstatusid	Char(1)		Estado civil
residentialphone	Char(50)		Telefone residencial
workphone:	Char(50)		Telefone comercial
cellphone	Char(50)		Telefone celular
messagephone	Char(50)		Telefone para recados
messagecontact	Char(50)		Contacto para recados
datebirth	date		Data de nascimento

cityidbirth	integer		Cidade de nascimento
countryidbirth	integer		Pais de nascimento
fatherid	integer		Pai
motherid	integer		Mãe
responsablelegalid	integer		Responsável legal
carplate	Char(40)		Placa do carro
specialnecessityid:	integer		Necessidade especial
specialnecessitydescription:	text		Descritivo da necessidade especial
cityidwork:	integer		Cidade do local de trabalho
zipcodework	Char(9)		CEP do local de trabalho
locationwork	Char(100)		Logradouro do local de trabalho
complementwork:	Char(40)		Complemento do local de trabalho
neighborhoodwork:	Char(100)		Bairro do local do trabalho
ethnicoriginid:	integer		Origem étnica
datedeath	date		Data de óbito do aluno
persontitleid	integer		
externalcourseidhs:	integer	FK	Código do curso de ensino médio
institutionidhs	integer	FK	Código da instituição do ensino médio
cityidhs	integer	FK	Código da cidade do ensino médio
yearhs:	integer		Ano de conclusão do ensino médio
isinsured:	boolean		E segurado
passive	Char(20)		Passivo

Tabela nº 29

Basphysicalpersonprofessor/ pessoa física - professor			
Campo	**Tipo**	**Chave**	**Descrição**
personid:	integer	PK	Código da pessoa
persondv	integer		Dígito verificador da pessoa
personmask:	Char(15)		Mascara para o Código da pessoa
name	Char(100)		Nome
shortname:	Char(30)		Apelido
cityid	integer		Código da cidade
zipcode	Char(9)		CEP
location:	Char(100)		Logradouro

number	Char(10)		Numero
complement	Char(60)		Complemento
neighborhood	text		Bairro
email	Char(60)		Email
emailalternative	Char(60)		Email alternativo
url:	Char(60)		URL
datein	date		Data de ingresso
password	Char(10)		Senha para acesso aos processo on-line
isallowpersonaldata	boolean		Permite a divulgação de dados pessoais
miolousername	Char(25)		nome do usuário que eh inserido automaticamente na base de dados do miolo para acesso ao modulo de serviços
sex	Char(1)		Sexo
maritalstatusid	Char(1)		Estado civil
residentialphone	Char(50)		Telefone residencial
workphone:	Char(50)		Telefone comercial
cellphone	Char(50)		Telefone celular
messagephone	Char(50)		Telefone para recados
messagecontact	Char(50)		Contacto para recados
datebirth	date		Data de nascimento
cityidbirth	integer		Cidade de nascimento
countryidbirth	integer		Pais de nascimento
fatherid	integer		Pai
motherid	integer		Mãe
responsablelegalid	integer		Responsável legal
carplate	Char(40)		Placa do carro
specialnecessityid:	integer		Necessidade especial
specialnecessitydescription:	text		Descritivo da necessidade especial
cityidwork:	integer		Cidade do local de trabalho
zipcodework	Char(9)		CEP do local de trabalho
locationwork	Char(100)		Logradouro do local de trabalho
complementwork:	Char(40)		Complemento do local de trabalho
neighborhoodwork:	Char(100)		Bairro do local do trabalho
ethnicoriginid:	integer		Origem étnica

datedeath	date		Data de óbito do aluno
persontitleid	integer		Título da pessoa

Tabela nº 30

Basprofessionalactivity/ actividades profissionais			
Campo	**Tipo**	**Chave**	**Descrição**
professionalactivityid:	integer	PK	Código da actividade profissional
description	text		Descrição

Tabela nº 31

Basprofessionalactivityagent/ agente da actividade profissional			
Campo	**Tipo**	**Chave**	**Descrição**
professionalactivityagentid:	integer	PK	Código do agente da actividade profissional
description:	text		Descrição

Tabela nº 32

Basprofessionalactivitylinktype/ tipos de vínculos nas actividades profissionais			
Campo	**Tipo**	**Chave**	**Descrição**
professionalactivitylinktypeid:	integer	PK	Código do tipo de vínculo das actividades profissionais
description:	text		Descrição
notifycompany:	boolean		Notificar a empresa no caso de fim do vínculo com a IES (estagio)

Tabela nº 33

Basprofessionalactivitypeople/ actividades profissionais das pessoas			
Campo	**Tipo**	**Chave**	**Descrição**
professionalactivitypeopleid	integer	PK	Código da actividade profissional da pessoa
professionalactivityid:	integer	FK	Código da actividade profissional
personid:	integer	FK	Código da pessoa
legalpersonid	integer	FK	Código da pessoa jurídica
begindate:	Date		Data de inicio
enddate:	Date		Data de término

professionalactivitylinktypeid	integer	FK	Código do tipo de vínculo da actividade profissional
professionalactivityagentid	integer	FK	Código do agente da actividade profissional (estagio)

Tabela nº 34

Bassector/ sectores			
Campo	**Tipo**	**Chave**	**Descrição**
sectorid	Integer	PK	Código do sector
description	Text		Descrição
email:	Char(60)		Email do sector
iscoordinatorssector	boolean		Se e um sector de coordenação

Tabela nº 35

Bassectorboss/ chefes de sectores			
Campo	**Tipo**	**Chave**	**Descrição**
bossid	Integer	PK,FK	Código do chefe de sector
sectorid:	Integer	PK,FK	Código do sector
level:	Integer		Nível
expirationlevel	interval		Nível de expiração
issendemail:	boolean		Envia/recebe emails
email:	Char(60)		Email

Tabela nº 36

Basspecialnecessity/ necessidades especiais			
Campo	**Tipo**	**Chave**	**Descrição**
specialnecessityid:	Integer	PK	Código da necessidade especial
description:	Text		Descrição
easyaccess:	boolean		Precisa de facilidades de acesso
accompanimentneeds:	boolean		Precisa de acompanhamento especial
ispermanent	boolean		E uma necessidade especial permanente
howmuchweeks:	Integer		Número de semanas da necessidade especial (no caso de não ser permanente)
begindate	date		Data inicial

Tabela nº 37

Basstamp/ carimbos			
Campo	**Tipo**	**Chave**	**Descrição**
stampid	Integer	PK	Código do carimbo
personid:	Integer	FK	Código da pessoa
functiondescription	text		Cargo
register:	text		Portaria

Tabela nº 38

Basstate/ estados/províncias da federação			
Campo	**Tipo**	**Chave**	**Descrição**
stateid	Char(2)	PK	Código do estado
countryid:	Integer	PK,FK	Código do país
name	Char(50)		Nome

Tabela nº 39

Basturn/ turnos			
Campo	**Tipo**	**Chave**	**Descrição**
turnid:	Integer	PK	Código do turno
description	Char(30)		Descrição
shortdescription:	Char(3)		Descrição sucinta
beginhour	time		Horário de inicio
endhou	time		Horário de término
charid:	Char(1)		Código de unificação de turnos

Tabela nº 40

Basunit/ unidades (campus)			
Campo	**Tipo**	**Chave**	**Descrição**
unitid	Integer	PK	Código da unidade
cityid:	Integer	FK	Código da cidade onde esta localizada a unidade
description:	Text		Descrição
accountingcode	Char(2)		Código contabilidade

Tabela nº 41

Basweekday/ dias da semana			
Campo	**Tipo**	**Chave**	**Descrição**
weekday	integer	PK	*Código do dia da semana*
description:	text		*Descrição*
shortdescription	Char(3)		*Descrição sucinta*

Anexo B - Guia para instalação e configuração do Sistema

A seguir são descritos os procedimentos necessários para a instalação do SAGU nas mais diversas distribuições GNU/Linux existentes e as configurações iniciais do sistema.

Por se tratar de uma aplicação *Web* o equipamento requerido, tanto para o seu desenvolvimento quanto para a utilização, não exige uma configuração especial, no entanto a utilização de um equipamento que será o servidor da aplicação terá uma configuração melhor referente à quantidade de memória.

Para a instalação do sistema SGU, os seguintes requisitos devem ser contemplados para o servidor:

Requisitos de software:

- Servidor web Apache 2.2
- Interpretador PHP 5.2.4
- Banco de dados PostgreSQL 8.2
- Xfvb 6.8.2
- OpenOffice.org 2.4
- Ubuntu 8.04

Requisitos de hardware:

Configuração mínima para funcionamento:

Processador Pentium IV ou similar

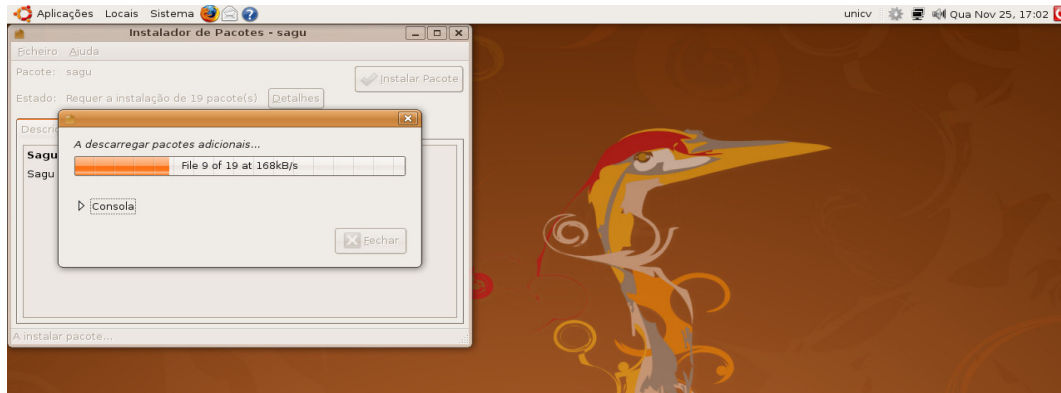
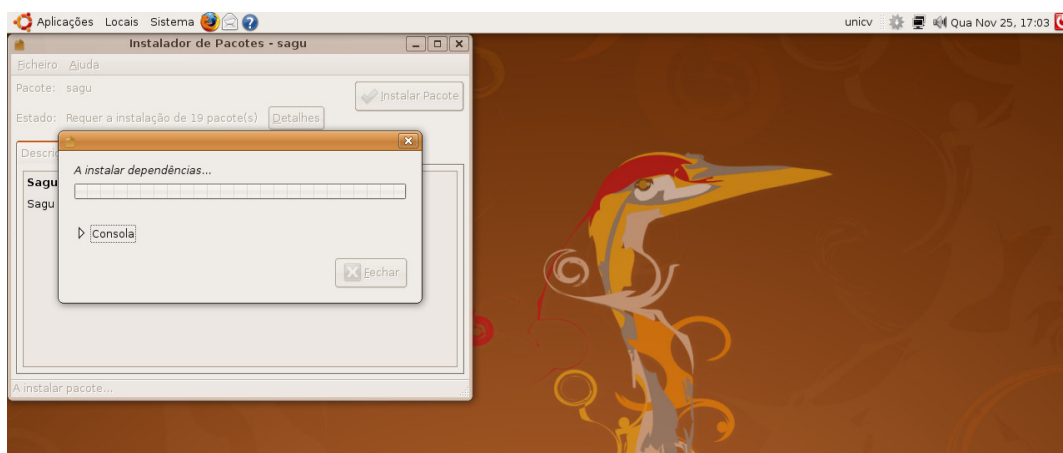
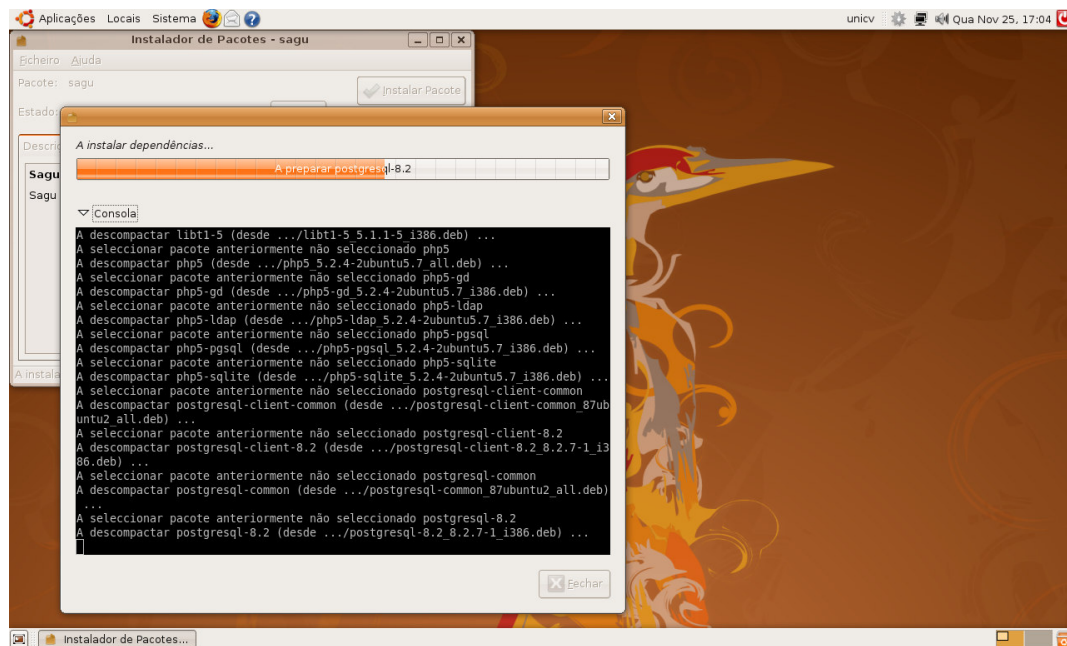
RAM de 512MB (1GB recomendado)

HD de 80GB (SATA recomendado)

Esta é a configuração mínima, geralmente utilizada apenas em ambiente de testes. Para utilização em ambiente de produção, recomenda-se um servidor de maior capacidade de

processamento e armazenamento. Tal servidor deve ser dimensionado de acordo com a quantidade prevista de utilizadores que terão acesso simultâneo ao sistema.

Instalação:

Para a instalação do SAGU, primeiro foi instalado o sistema operativo *Ubuntu* 8.04 e feito actualizações de pacotes.

A instalação pode ser efectuada em modo de consola através de comandos ou em modo interface. Neste trabalho o sistema foi instalado no modo interface através do **menu sistema -> gestor de pacotes synaptic**, que permite a instalação e actualização de todos os pacotes de softwares.

O SAGU requer a instalação, prévia do Apache, PostgreSQL e PHP, que já estão embutidos no pacote de instalação do SAGU2 que foi instalado.

Configurações

Tradução

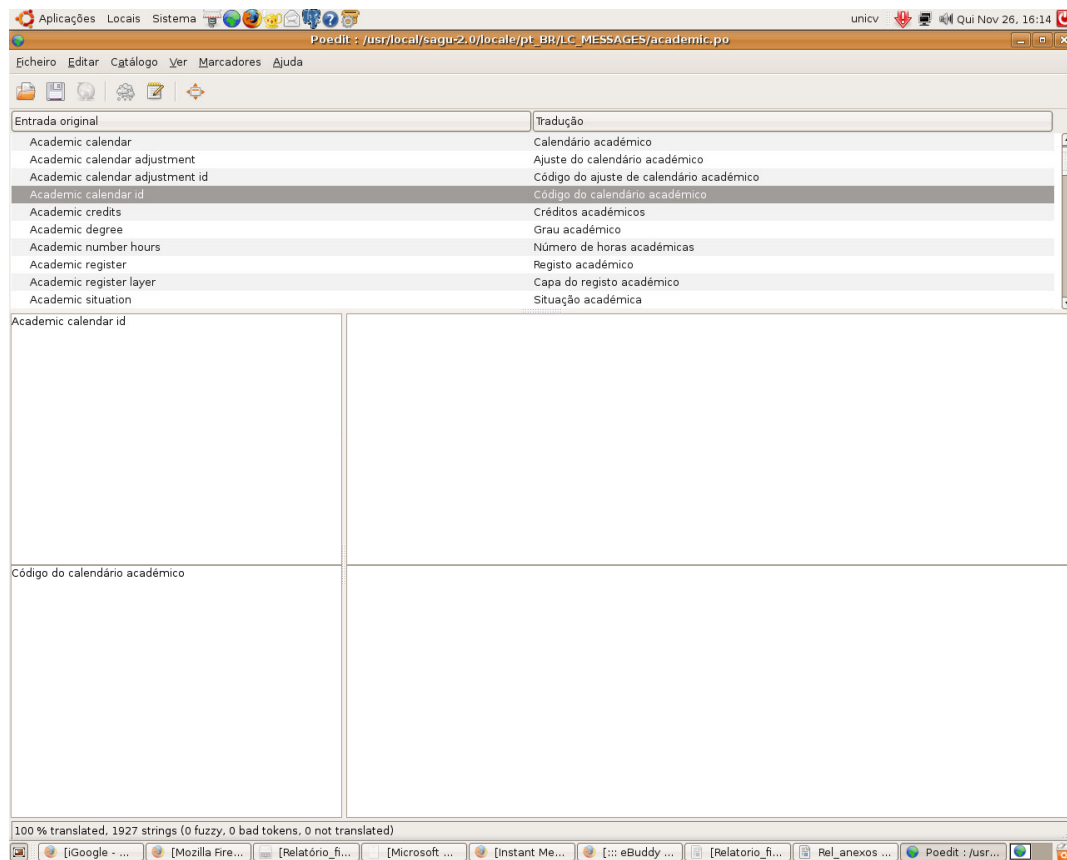
Após a instalação do sistema, foi feita a tradução o sistema do Inglês para português. Assim foram efectuadas as seguintes configurações.

É necessário adicionar a codificação "ISO-8859-1" e gerar novamente os locales do sistema. Para isso faça:

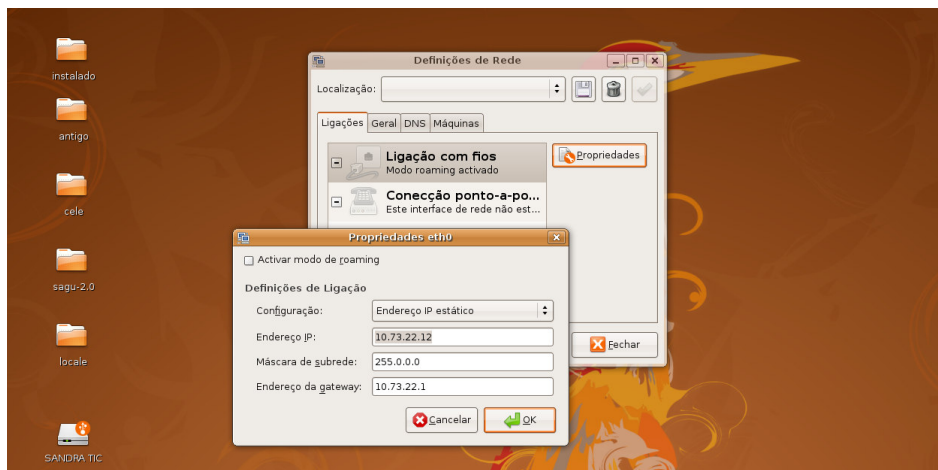
echo "pt_BR.ISO-8859-1 ISO-8859-1" >> /var/lib/locales/supported.d/pt

locale-gen

localedef -i pt_BR -c -f ISO-8859-1 pt_BR

Para além disso, também foi utilizado a ferramenta Poedit para uma melhor correcção de português,

Colocação do sistema na rede

Após a instalação e a tradução foi colocado o sistema na rede. Para isso procedeu-se o seguinte:

Configuração do endereço IP Estático;

Configuração do ficheiro sagu2 no directório /etc/apache2/sites-available;

Para efectuar esta configuração criou-se um VirtualHost e ServerName com o endereço IP que foi configurado estaticamente no servidor.

```
<VirtualHost 10.73.22.71:80>
  ServerAdmin sagu2@solis.coop.br
  DocumentRoot /usr/local/sagu-2.0/html
  ServerName 10.73.22.71/
</VirtualHost>
<Directory "/usr/local/sagu-2.0/html">
    AllowOverride None
    Options FollowSymLinks
    Order allow,deny
    Allow from all
</Directory>
```

Como foi criado um VirtualHost diferente de "sagu2local", foi configurado o arquivo /usr/local/sagu-2.0/etc/miolo.conf.

Configuração do ficheiro miolo.conf do SAGU2;

Para completar a configuração, foi introduzido o endereço IP configurado anteriormente no ficheiro miolo.conf, para que possamos ter acesso ao sistema através da URL.

Parte do código utilizado:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<conf>
<home>
      <miolo>/usr/local/sagu-2.0</miolo>
```

```
    <classes>/usr/local/sagu-2.0/classes</classes>
    <modules>/usr/local/sagu-2.0/modules</modules>
    <etc>/usr/local/sagu-2.0/etc</etc>
    <logs>/usr/local/sagu-2.0/var/log</logs>
    <trace>/usr/local/sagu-2.0/var/trace</trace>
    <db>/usr/local/sagu-2.0/var/db</db>
    <html>/usr/local/sagu-2.0/html</html>
    <themes>/usr/local/sagu-2.0/classes/ui/themes</themes>
    <extensions>/usr/local/sagu-2.0/extensions</extensions>
    <reports>/usr/local/sagu-2.0/var/reports</reports>
    <images>/usr/local/sagu-2.0/ui/images</images>
    <url>http://10.73.22.71</url>
    <url_themes>/themes</url_themes>
    <url_reports>/reports</url_reports>
    <module.themes>/ui/themes</module.themes>
    <module.html>/html</module.html>
    <module.images>/html/images</module.images>
  </home>
```